(12) UK Patent Application (19) GB (11) 2 344 197 (13) A

(43) Date of A Publication 31.05.2000

(21) Application No 9918517.5

(22) Date of Filing 05.08.1999

(30) Priority Data
(31) 9129316 (32) 05.08.1998 (33) US

(51) INT CL[7]
G06F 17/30

(52) UK CL (Edition R)
G4A AUDB

(56) Documents Cited
GB 2339374 A GB 2329310 A EP 0947935 A2
EP 0926610 A1 EP 0778534 A1 WO 99/57657 A1
WO 99/35802 A2 WO 98/45793 A1 WO 97/18635 A2
WO 97/18516 A1 WO 97/17662 A1 US 5742762 A

(58) Field of Search
UK CL (Edition R) G4A AFD AUDB
INT CL[7] G06F 17/21 17/30
ONLINE: COMPUTER EPODOC JAPIO WPI INTERNET

(71) Applicant(s)
Spyglass Inc
(Incorporated in USA - Delaware)
1240 E Diehl, Naperville, Illinois 60563,
United States of America

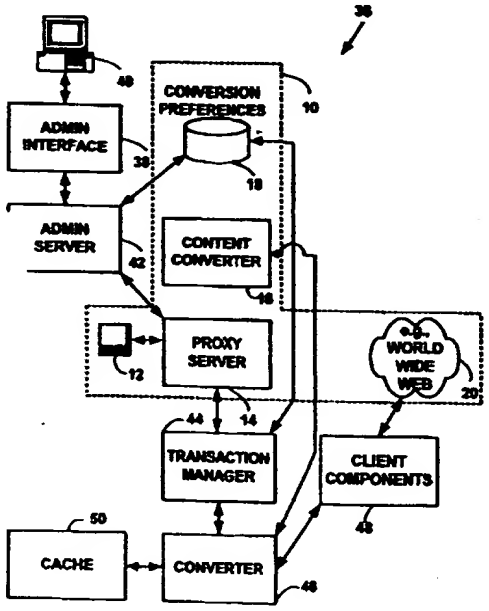
(72) Inventor(s)
Kenneth F Carlino

(74) Agent and/or Address for Service
Frank B Dehn & Co
179 Queen Victoria Street, LONDON, EC4V 4EL,
United Kingdom

(54) Abstract Title
Content conversion of electronic documents

(57) Original electronic documents such as a web page in a standard format (e.g., SuperVGA, 800x600 pixel resolution, 256 colors from the Internet or an intranet are converted into converted formats that can be appropriately displayed on a hand-held device or other desired device with a display with less than SuperVGA resolution. However, converted electronic documents can also be displayed on the display with SuperVGA resolution. A conversion directive is used to convert an original electronic document into a converted electronic document. It includes a list of conversion preferences and uses conversion precedence indicating an ordering in which the list of conversion preferences are to be applied to the original electronic document. The conversion preferences include user-conversion preferences, device-conversion preferences, site-specific conversion preferences, or other preferences. The content converter can work with a proxy server.

FIG. 3



GB 2 344 197 A

FIG. 1

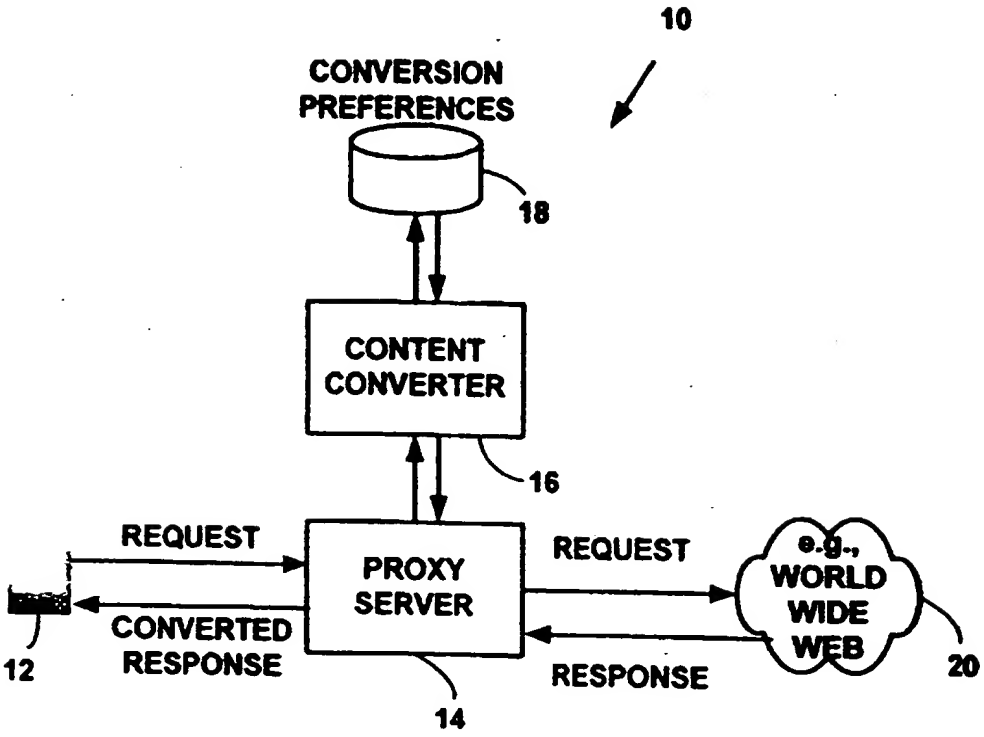

10
CONVERSION PREFERENCES
18
CONTENT CONVERTER
16
REQUEST
PROXY SERVER
REQUEST
e.g., WORLD WIDE WEB
CONVERTED RESPONSE
RESPONSE
12
14
20

FIG. 2

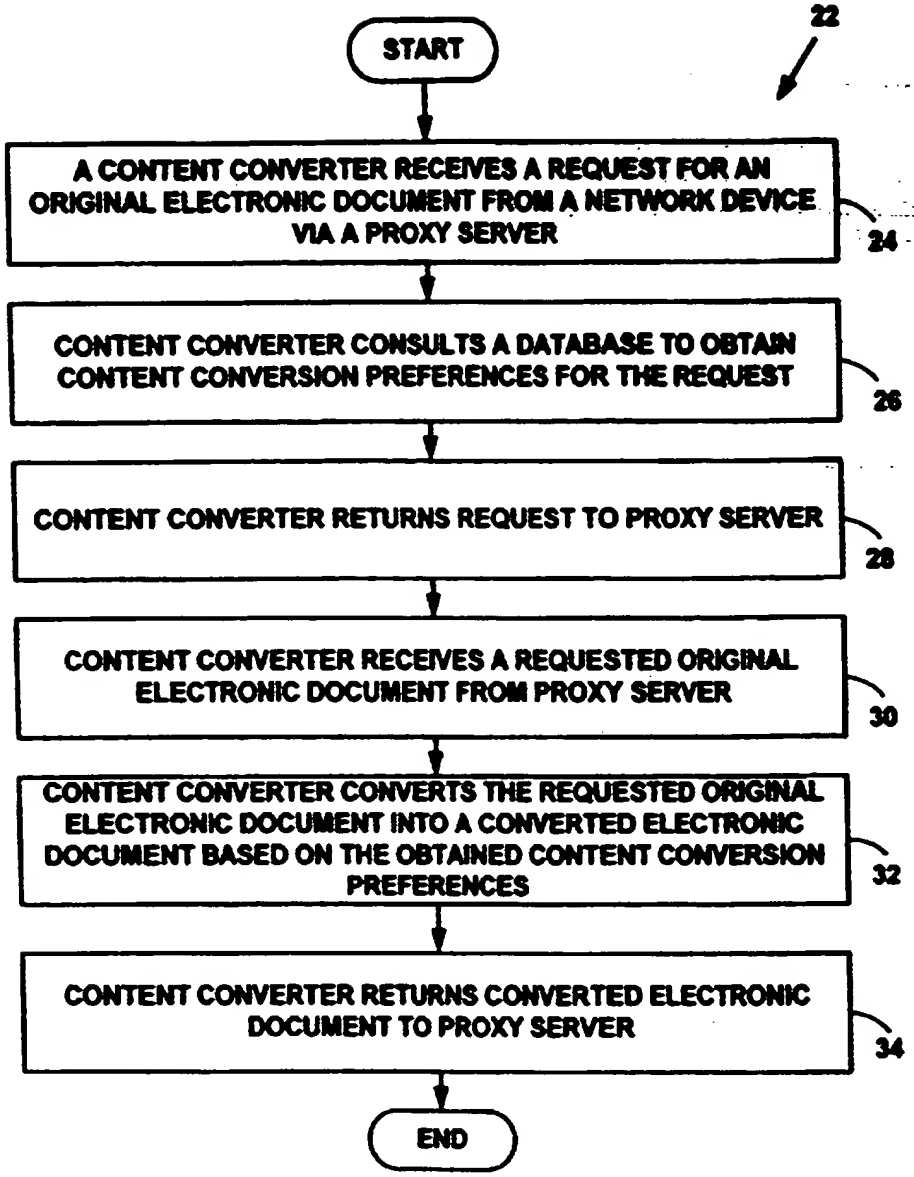

22
START
A CONTENT CONVERTER RECEIVES A REQUEST FOR AN ORIGINAL ELECTRONIC DOCUMENT FROM A NETWORK DEVICE VIA A PROXY SERVER
24
CONTENT CONVERTER CONSULTS A DATABASE TO OBTAIN CONTENT CONVERSION PREFERENCES FOR THE REQUEST
26
CONTENT CONVERTER RETURNS REQUEST TO PROXY SERVER
28
CONTENT CONVERTER RECEIVES A REQUESTED ORIGINAL ELECTRONIC DOCUMENT FROM PROXY SERVER
30
CONTENT CONVERTER CONVERTS THE REQUESTED ORIGINAL ELECTRONIC DOCUMENT INTO A CONVERTED ELECTRONIC DOCUMENT BASED ON THE OBTAINED CONTENT CONVERSION PREFERENCES
32
CONTENT CONVERTER RETURNS CONVERTED ELECTRONIC DOCUMENT TO PROXY SERVER
34
END

FIG. 3

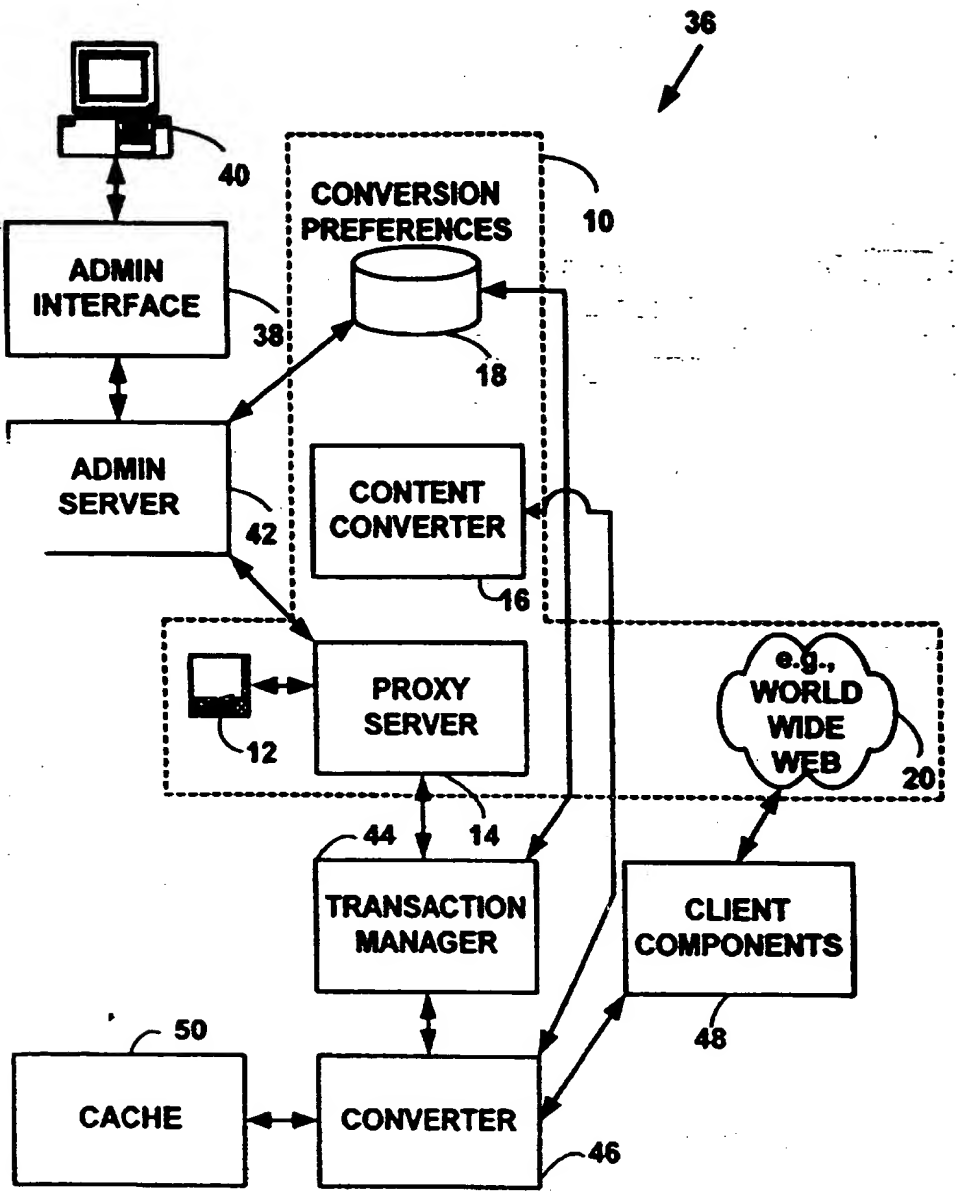

36
40
CONVERSION PREFERENCES
10
ADMIN INTERFACE
38
18
ADMIN SERVER
42
CONTENT CONVERTER
16
PROXY SERVER
12
e.g., WORLD WIDE WEB
20
44
14
TRANSACTION MANAGER
CLIENT COMPONENTS
48
50
CACHE
CONVERTER
46

FIG. 4A

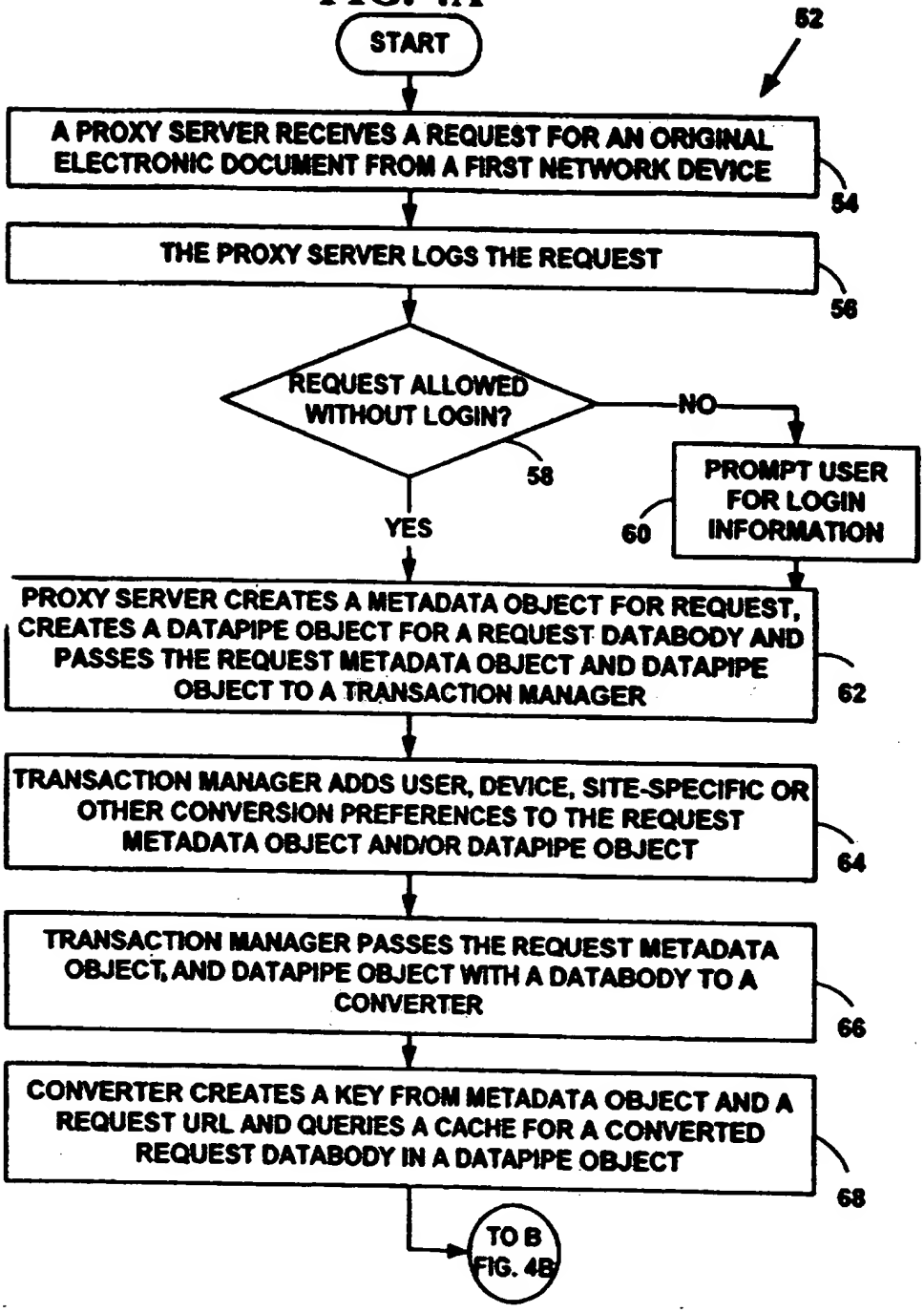

START
52
A PROXY SERVER RECEIVES A REQUEST FOR AN ORIGINAL ELECTRONIC DOCUMENT FROM A FIRST NETWORK DEVICE
54
THE PROXY SERVER LOGS THE REQUEST
56
REQUEST ALLOWED WITHOUT LOGIN?
58
NO
YES
PROMPT USER FOR LOGIN INFORMATION
60
PROXY SERVER CREATES A METADATA OBJECT FOR REQUEST, CREATES A DATAPIPE OBJECT FOR A REQUEST DATABODY AND PASSES THE REQUEST METADATA OBJECT AND DATAPIPE OBJECT TO A TRANSACTION MANAGER
62
TRANSACTION MANAGER ADDS USER, DEVICE, SITE-SPECIFIC OR OTHER CONVERSION PREFERENCES TO THE REQUEST METADATA OBJECT AND/OR DATAPIPE OBJECT
64
TRANSACTION MANAGER PASSES THE REQUEST METADATA OBJECT, AND DATAPIPE OBJECT WITH A DATABODY TO A CONVERTER
66
CONVERTER CREATES A KEY FROM METADATA OBJECT AND A REQUEST URL AND QUERIES A CACHE FOR A CONVERTED REQUEST DATABODY IN A DATAPIPE OBJECT
68
TO B FIG. 4B

FIG. 4B

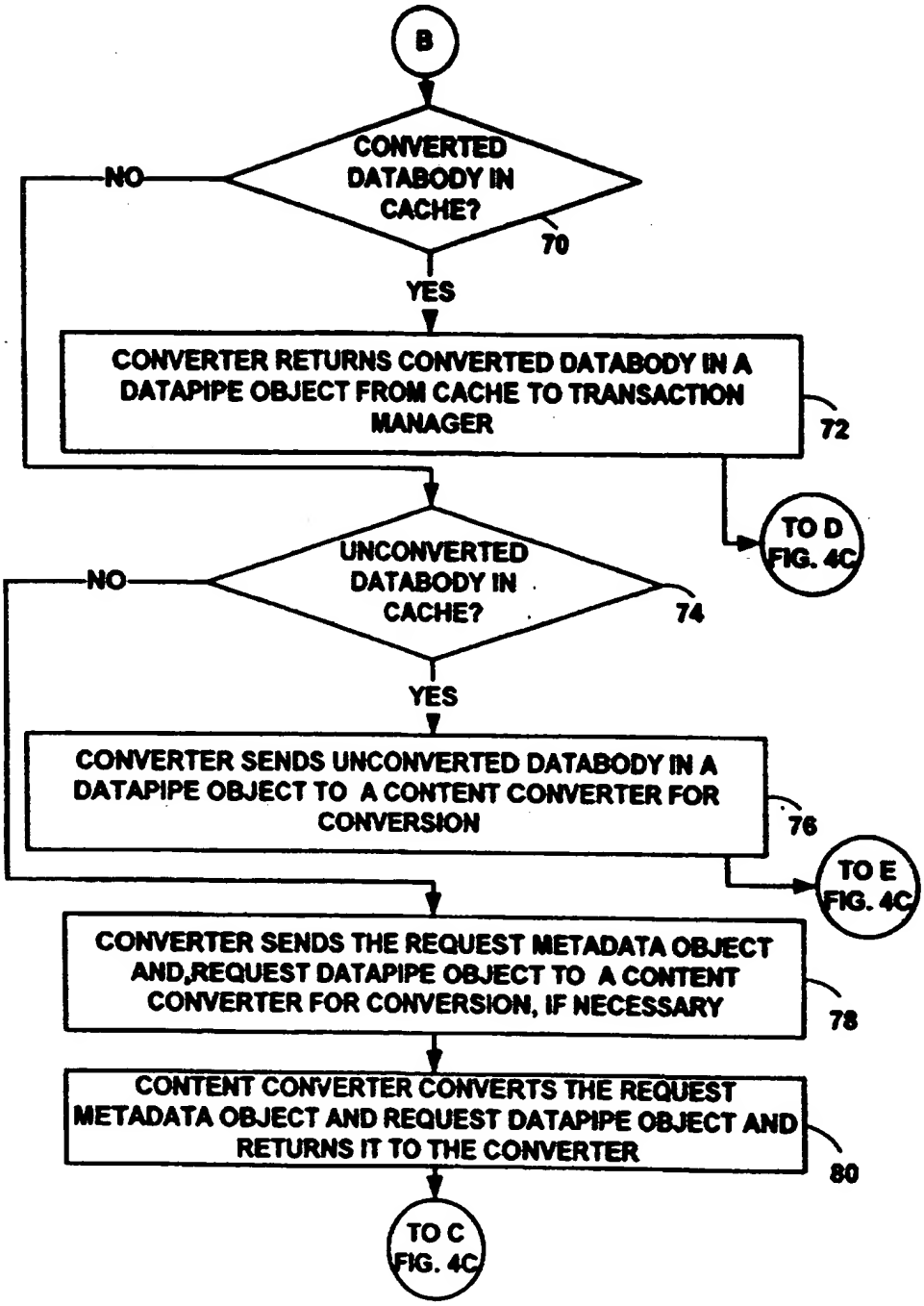

B
CONVERTED DATABODY IN CACHE?
70
NO
YES
CONVERTER RETURNS CONVERTED DATABODY IN A DATAPIPE OBJECT FROM CACHE TO TRANSACTION MANAGER
72
TO D FIG. 4C
UNCONVERTED DATABODY IN CACHE?
74
NO
YES
CONVERTER SENDS UNCONVERTED DATABODY IN A DATAPIPE OBJECT TO A CONTENT CONVERTER FOR CONVERSION
76
TO E FIG. 4C
CONVERTER SENDS THE REQUEST METADATA OBJECT AND,REQUEST DATAPIPE OBJECT TO A CONTENT CONVERTER FOR CONVERSION, IF NECESSARY
78
CONTENT CONVERTER CONVERTS THE REQUEST METADATA OBJECT AND REQUEST DATAPIPE OBJECT AND RETURNS IT TO THE CONVERTER
80
TO C FIG. 4C

FIG. 4C

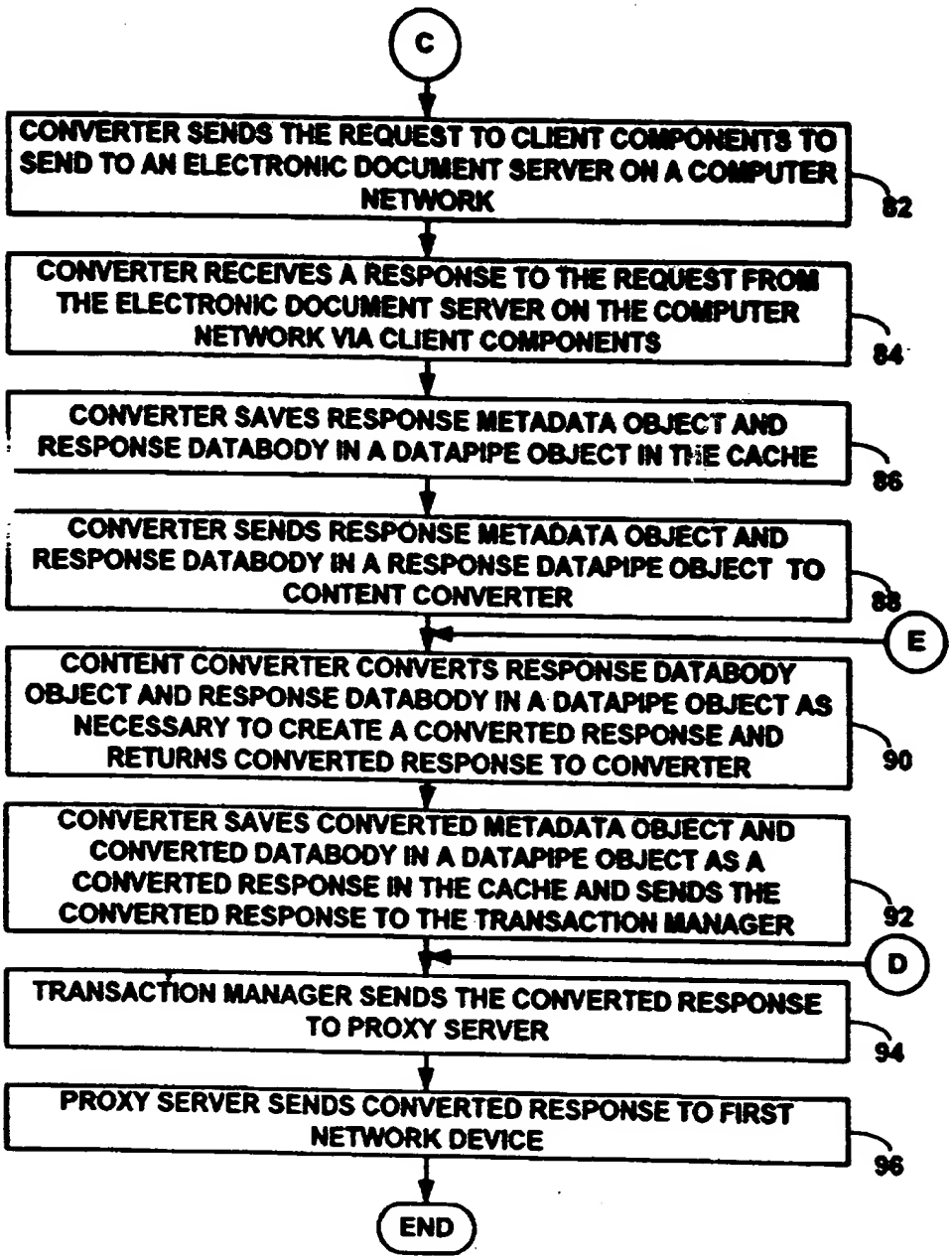

C
CONVERTER SENDS THE REQUEST TO CLIENT COMPONENTS TO SEND TO AN ELECTRONIC DOCUMENT SERVER ON A COMPUTER NETWORK
82
CONVERTER RECEIVES A RESPONSE TO THE REQUEST FROM THE ELECTRONIC DOCUMENT SERVER ON THE COMPUTER NETWORK VIA CLIENT COMPONENTS
84
CONVERTER SAVES RESPONSE METADATA OBJECT AND RESPONSE DATABODY IN A DATAPIPE OBJECT IN THE CACHE
86
CONVERTER SENDS RESPONSE METADATA OBJECT AND RESPONSE DATABODY IN A RESPONSE DATAPIPE OBJECT TO CONTENT CONVERTER
88
E
CONTENT CONVERTER CONVERTS RESPONSE DATABODY OBJECT AND RESPONSE DATABODY IN A DATAPIPE OBJECT AS NECESSARY TO CREATE A CONVERTED RESPONSE AND RETURNS CONVERTED RESPONSE TO CONVERTER
90
CONVERTER SAVES CONVERTED METADATA OBJECT AND CONVERTED DATABODY IN A DATAPIPE OBJECT AS A CONVERTED RESPONSE IN THE CACHE AND SENDS THE CONVERTED RESPONSE TO THE TRANSACTION MANAGER
92
D
TRANSACTION MANAGER SENDS THE CONVERTED RESPONSE TO PROXY SERVER
94
PROXY SERVER SENDS CONVERTED RESPONSE TO FIRST NETWORK DEVICE
96
END

FIG. 5

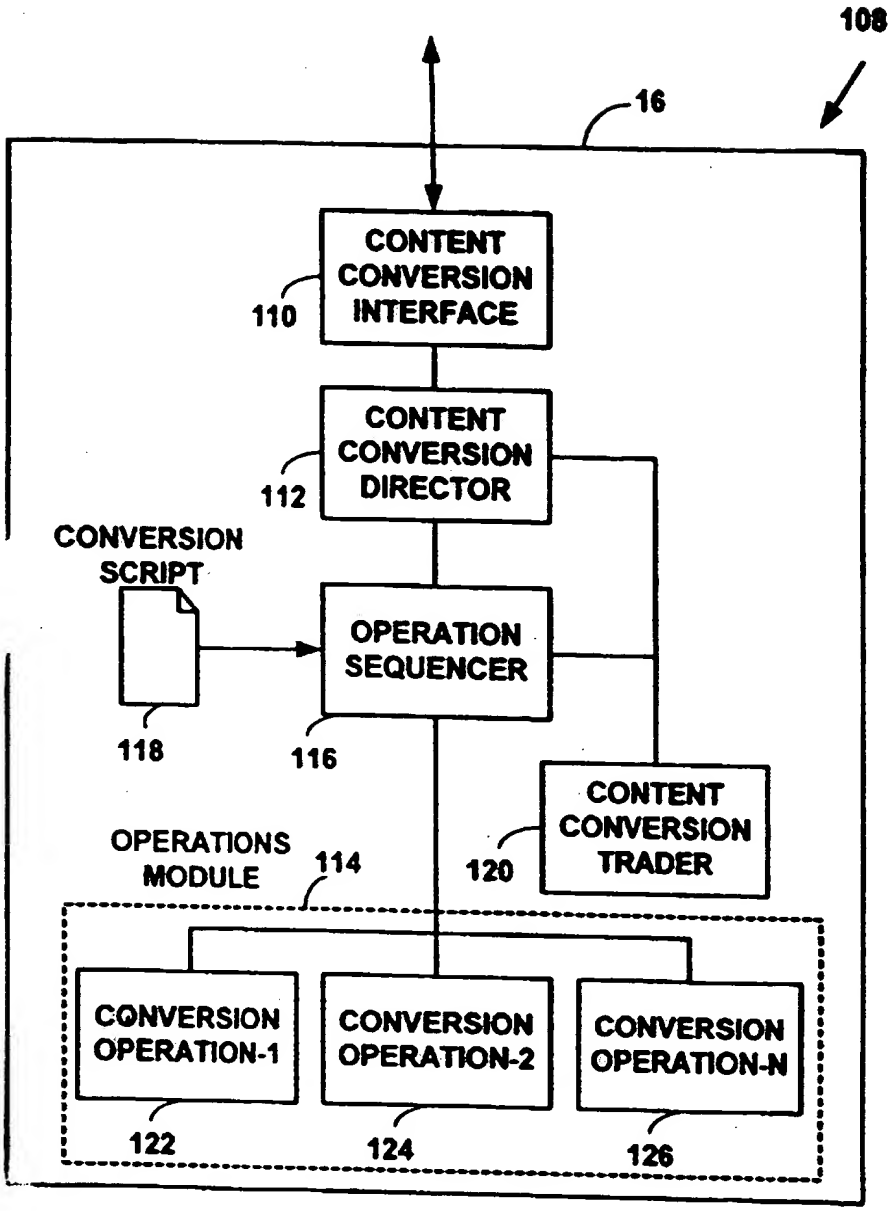

108
16
CONTENT CONVERSION INTERFACE
110
CONTENT CONVERSION DIRECTOR
112
CONVERSION SCRIPT
118
OPERATION SEQUENCER
116
CONTENT CONVERSION TRADER
120
OPERATIONS MODULE
114
CONVERSION OPERATION-1
122
CONVERSION OPERATION-2
124
CONVERSION OPERATION-N
126

FIG. 6

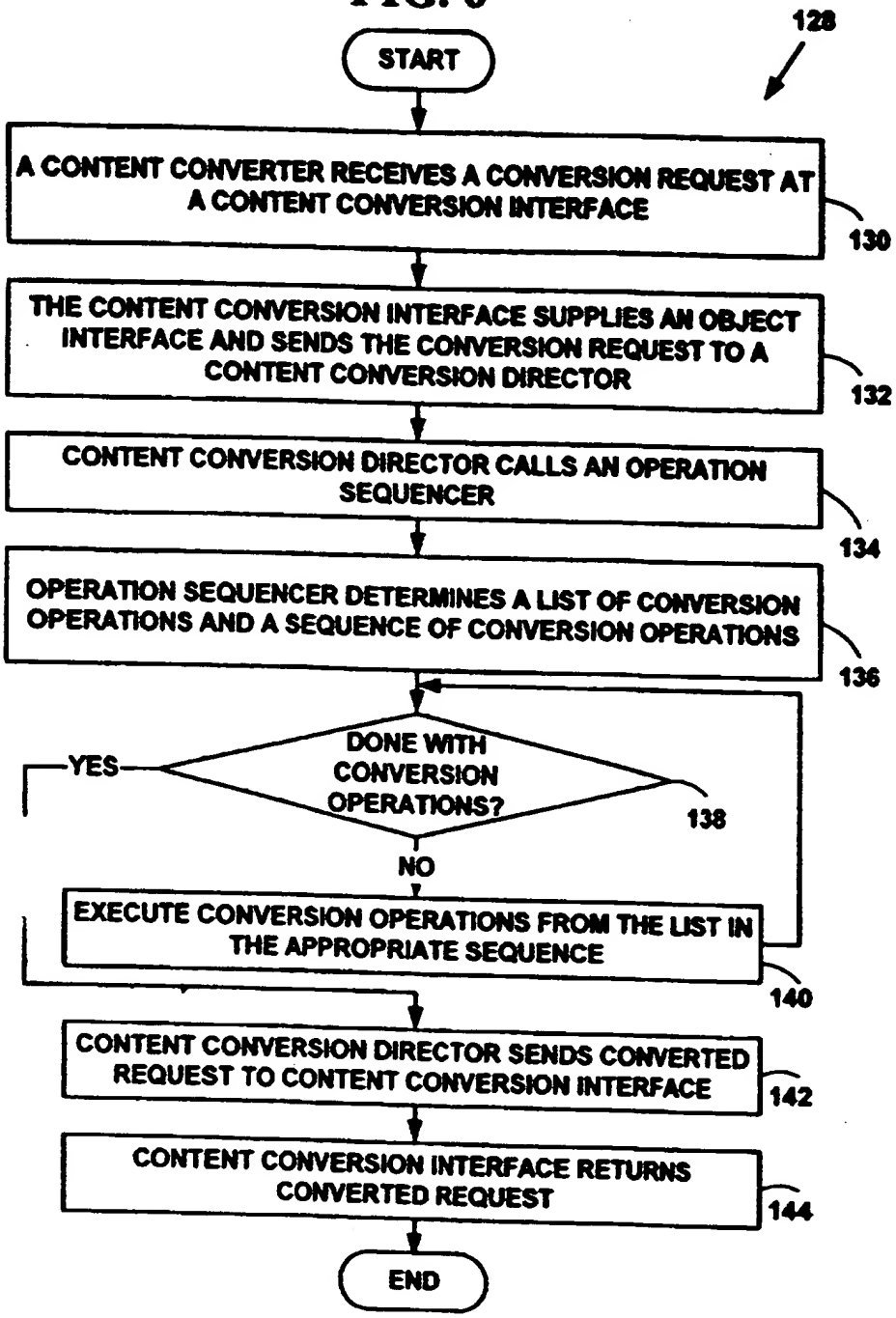

128
START
A CONTENT CONVERTER RECEIVES A CONVERSION REQUEST AT A CONTENT CONVERSION INTERFACE
130
THE CONTENT CONVERSION INTERFACE SUPPLIES AN OBJECT INTERFACE AND SENDS THE CONVERSION REQUEST TO A CONTENT CONVERSION DIRECTOR
132
CONTENT CONVERSION DIRECTOR CALLS AN OPERATION SEQUENCER
134
OPERATION SEQUENCER DETERMINES A LIST OF CONVERSION OPERATIONS AND A SEQUENCE OF CONVERSION OPERATIONS
136
DONE WITH CONVERSION OPERATIONS?
138
YES
NO
EXECUTE CONVERSION OPERATIONS FROM THE LIST IN THE APPROPRIATE SEQUENCE
140
CONTENT CONVERSION DIRECTOR SENDS CONVERTED REQUEST TO CONTENT CONVERSION INTERFACE
142
CONTENT CONVERSION INTERFACE RETURNS CONVERTED REQUEST
144
END

FIG. 7

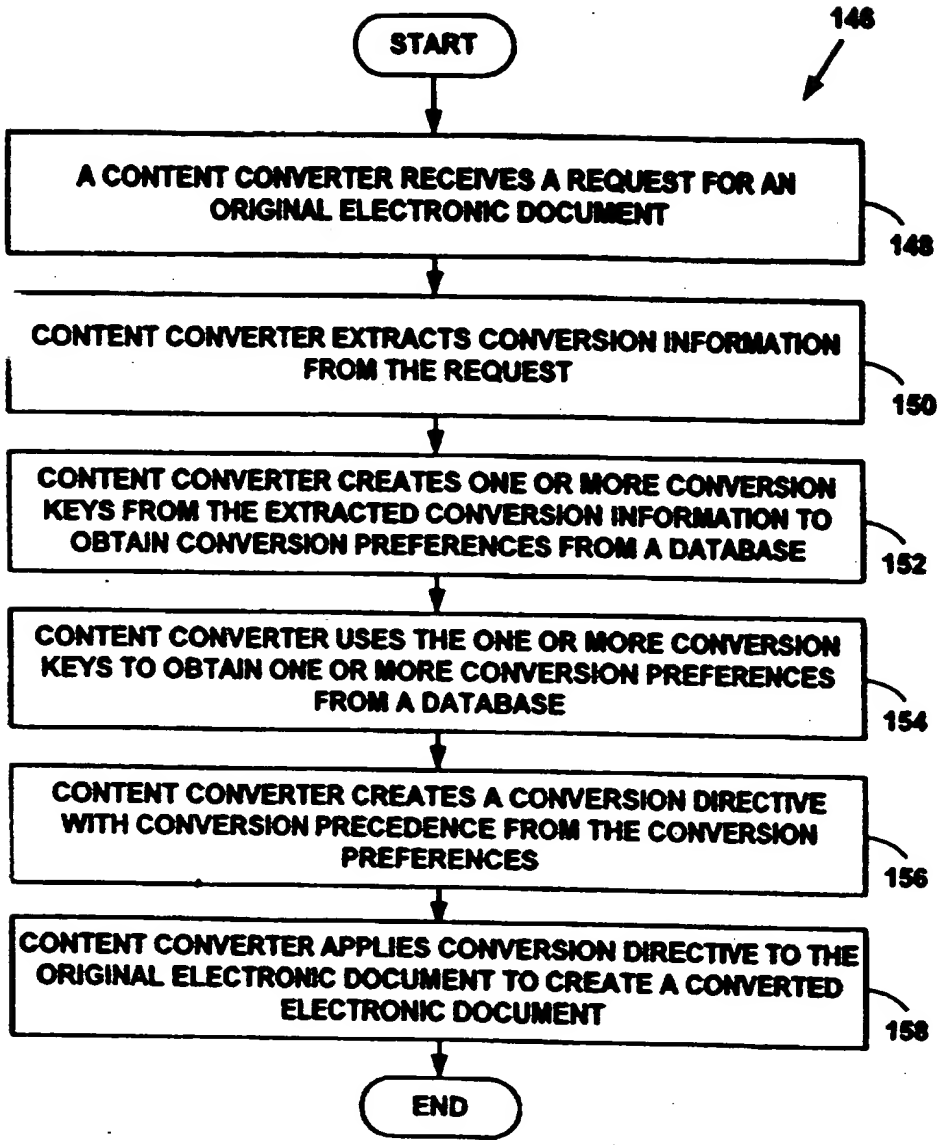
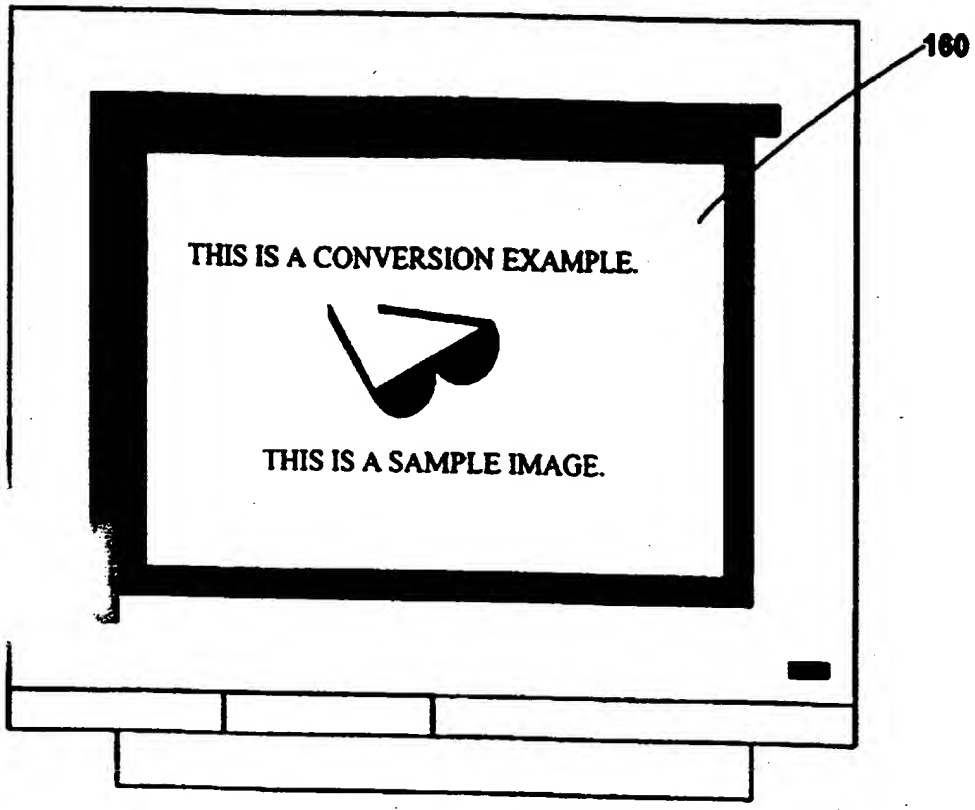
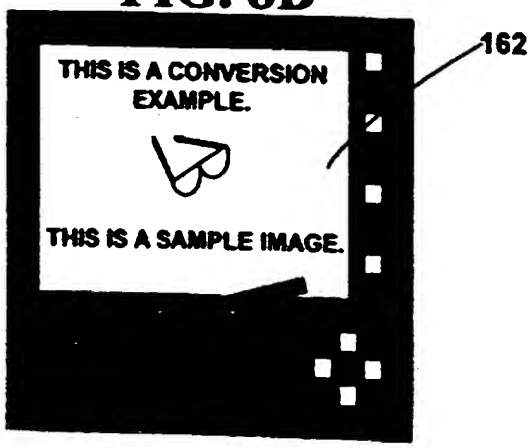

146
START
A CONTENT CONVERTER RECEIVES A REQUEST FOR AN ORIGINAL ELECTRONIC DOCUMENT
148
CONTENT CONVERTER EXTRACTS CONVERSION INFORMATION FROM THE REQUEST
150
CONTENT CONVERTER CREATES ONE OR MORE CONVERSION KEYS FROM THE EXTRACTED CONVERSION INFORMATION TO OBTAIN CONVERSION PREFERENCES FROM A DATABASE
152
CONTENT CONVERTER USES THE ONE OR MORE CONVERSION KEYS TO OBTAIN ONE OR MORE CONVERSION PREFERENCES FROM A DATABASE
154
CONTENT CONVERTER CREATES A CONVERSION DIRECTIVE WITH CONVERSION PRECEDENCE FROM THE CONVERSION PREFERENCES
156
CONTENT CONVERTER APPLIES CONVERSION DIRECTIVE TO THE ORIGINAL ELECTRONIC DOCUMENT TO CREATE A CONVERTED ELECTRONIC DOCUMENT
158
END

FIG. 8A



SUPERVGA RESOLUTION DISPLAY

FIG. 8B



PDA GRAYSCALE DISPLAY

Content Conversion of Electronic Data

This invention relates to computer networks. More specifically, it relates to a method and system for converting the content of electronic data for a desired network device using conversion preferences.

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by any person of the patent disclosure, as it appears in the Patent Office records and documents. In all other respects whatsoever copyright is reserved.

The Internet is a world-wide network of interconnected computers. The World-Wide-Web is an information system on the Internet designed for electronic document interchange. Electronic documents on the World-Wide-Web are typically stored in files that include text, hypertext, references to graphics, animation, audio, video and other electronic data. The structure of hypertext documents is defined by document markup languages such as Standard Generalized Markup Language ("SGML"), Hyper Text Markup Language ("HTML"), eXtensible Markup Language ("XML"), Virtual Reality Markup Language ("VRML") and others.

As is known in the art, a hypertext document includes markup codes called "tags." Tags define the structure of a hypertext document and typically includes at least a "begin" tag name enclosed by a delimiter and, in many instances, an "end" tag name enclosed by a delimiter. For example, the markup tag "<H1>" signifies the beginning of a Hyper Text Markup Language first level header, and the markup tag "</H1>" signifies the end of a Hyper Text Markup Language first level header. However, the Hyper Text Markup Language

image tag "<IMG ...>" ends with the closing tag delimiter ">" and does not use an end tag in the format "<\IMG>". Other markup languages have similar tags used to create hypertext documents.

Markup languages allow references to additional content besides text including graphics, animation, audio, video and other electronic data. The Hyper Text Markup Language allows use of graphical images in a hypertext document with an image "<IMG>" tag. For example, an exemplary Hyper Text Markup Language image tag <IMG SRC="logo.jpg"> allows a graphical logo image stored in a Joint Pictures Expert Group file "logo.jpg" to be displayed.

Hypertext documents from the World-Wide-Web are typically displayed for a user with a software application called a "browser" such as Internet Explorer, by Microsoft Corporation of Redmond Washington, or Netscape Navigator, by Netscape Communications of Mountain View, California, and others. A browser typically parses a hypertext document and converts hypertext, including markup tags, into a visual display of text, graphics, animation, audio, video, etc., for display on a device such as a personal computer display.

Additional content is retrieved in a hypertext document from other sources using "hyperlink" references within hypertext documents. For example, an exemplary Hyper Text Markup Language hyperlink tag "<A HREF="http://www.spyglass.com/logo.mov">" provides a hyperlink to a movie file "logo.mov." When a user selects the link (e.g., with a mouse click) in a hypertext document, the movie file "logo.mov" is located using a Uniform Resource Locator ("URL") from the location "www.spyglass.com." Hyper Text Transfer Protocol (e.g., "HTTP") is used as the transfer protocol.

Transfer protocols such as Hyper Text Transfer Protocol ("HTTP"), File Transfer Protocol ("FTP"), Gopher, and others provide a means for transferring hypertext documents or additional content from other locations on the World-Wide-Web. Hyper Text Transfer Protocol is one primary protocol used to transfer information on the World-Wide-Web. Hyper Text Transfer Protocol is a protocol that allows users to connect to a server, make a hypertext request, get a response, and then disconnect from the server. File Transfer Protocol is a protocol that provides access to files on remote systems. Using File Transfer Protocol, a user logs onto a system, searches a directory structure and downloads or uploads a file. Gopher is a protocol similar to File Transfer Protocol. Gopher provides a series of menus linked to files containing actual hypertext.

Content providers on the World-Wide-Web provide custom content using attributes from markup language tags. For example, the Hyper Text Markup Language IMG tag includes the following attributes: ISMAP, a selectable image map; SRC, a source Uniform Resource Locator of an image; ALT, a text string used instead of an image; ALIGN, for alignment of an image (e.g., left, middle, right); VSPACE, the space between an image and the text above and below it; HSPACE, the space between and image and the text to its left or right; WIDTH, the width in pixels of an image; HEIGHT, the height in pixels of an image; and a few other attributes depending on the browser being used (e.g., BORDER and LOWSRC in a Netscape browser). In addition, other content attributes can be modified such as one of 256 colors (e.g., TEXT="blue," or TEXT="0xa6caf0" for sky blue), font face (e.g., FONT FACE="Times Roman"), character formatting, (e.g., <B>text</B> for bold text), etc.

Most of the electronic documents developed for the World-Wide-Web assume that users will view the content of the electronic document with a browser on a desktop computer screen with a standard "SuperVGA" resolution (e.g., 800x600 pixel resolution with 256 or

more available colors). A user can alter display of selected electronic document content by changing browser attributes (e.g., color of text, size of text). However, the changeable content is limited by changeable attributes in a browser being used.

There are a number of problems with using electronic documents developed for the World-Wide-Web based on the assumption of viewing with standard "SuperVGA" resolution. A user with a hand-held computer, personal digital assistant ("PDA") or other hand-held device, such as wireless phone, may desire to view electronic documents from the World-Wide-Web. However, most hand-held devices have a display with a resolution that is less than SuperVGA (e.g., less than 800x600 pixel resolution and less than 256 colors) and typically do not support color. Thus, the content of electronic documents will not be properly displayed and may be difficult to view based on the original hypertext content.

One solution is to store several versions of an electronic document for different devices. However, this often makes administration of the electronic documents very difficult. Multiple copies of electronic documents also waste a tremendous amount of storage space on electronic document servers on the World-Wide-Web. The dynamic nature of the World-Wide-Web also means that new content types are constantly being introduced. Multiple copies of existing electronic documents must be constantly manipulated.

Another solution is to provide content conversion applications to convert electronic documents to match the capabilities of a device or the preferences of a user. One purpose of converting electronic documents is to provide information to users in a format different from that provided by the initial electronic document provider. Examples include scaling images for a specific device with a small display or converting text to speech for a visually impaired user. Most content conversion applications depend on a user-device to render the electronic

document being displayed. As a result, each hand-held device is required to have a copy of a desired content conversion application.

There have been attempts to provide content conversion applications for electronic documents for the World-Wide-Web. QuickWeb Technology, by Intel Corporation of Santa Clara, California, analyzes a hypertext document for graphic images and then compresses bits of image data so that the images can be transmitted faster to a user's device. QuickWeb Technology also caches images. After a user's first request, an image is cached by QuickWeb. Subsequent requests for the same image are delivered from the cached source instead of a remote content server. QuickWeb Technology is limited to converting images and currently does not allow conversion of other content information.

TranSend, by the University of California at Berkeley, California, provides a conversion proxy that converts images in an electronic document by reducing image quality. Although the resulting images are of a lower quality, an original image can also be recovered. The TranSend Proxy distills images for faster electronic document display from the World-Wide-Web. TranSend limits user configurations to turning the proxy on and off and reducing image quality for faster transmission. TranSend is also currently limited to converting images only and does not allow conversion of other information.

Thus, it is desirable to have a content conversion application that converts other aspects of electronic documents besides images for display on a desired device. The content conversion application should allow conversion of an electronic document based on capabilities of a device, display preferences of a user, site-specific characteristics of an electronic document server, or other preferences.

Viewed from one aspect of the present invention there is provided a method of content conversion in a first computer network with a plurality of network devices connected

to a second computer network with a plurality of network devices, comprising the following steps:

receiving a request for an original electronic document on a second network device on the first computer network from a first network device on the first computer network;

extracting conversion information from the request;

creating one or more conversion keys from the extracted conversion information to access one or more of a plurality of conversion preferences stored in a database;

obtaining one or more of the plurality of conversion preferences from the database using the one or more conversion keys; and

creating a conversion directive with a conversion precedence to convert an original electronic document into a converted electronic document.

Viewed from another aspect of the present invention there is provided a method of content conversion in a first computer network with a plurality of network devices connected to a second computer network with a plurality of network devices, comprising the following steps:

receiving an original electronic document on a second network device on the first network from a third network device on the second network;

obtaining a conversion directive with a conversion precedence on the second network device, wherein the conversion directive includes a list of conversion preferences and uses conversion precedence indicating an ordering in which the list of conversion preferences are to be applied to the original electronic document; and

converting the original electronic document into a converted electronic document using the conversion directive.

Viewed from another aspect of the present invention there is provided computer software which when run on data processing means will cause the above methods to be carried out. The software may be provided on a computer readable medium such as a disk or as signals from a remote site, for example.

Viewed from another aspect of the present invention, there is provided a content conversion system comprising, in combination:

a content converter for converting an original electronic document into a converted electronic document for a desired device based on a content conversion directive;

a database for storing a plurality of conversion preferences used for content conversion;

a proxy server for communicating with a plurality of network devices making requests for original electronic documents, communicating with the context converter, and with a plurality of content servers on a computer network providing original electronic documents; and

a content conversion directive with a conversion precedence created from one or more of the plurality of conversion preferences from the database for converting an original electronic document into a converted electronic document.

Viewed from another aspect there is provided computer software which when run on data processing means will cause the data processing means to operate as such a system.

In accordance with a preferred embodiment of the present invention, some of the problems associated with content conversion of electronic documents are overcome. A method and system for content conversion of electronic documents using conversion preferences is provided. The method and system allow content conversion of an original electronic document including text, hypertext, images and other electronic document content based on one or more sets of conversion preferences.

A content conversion method for a preferred embodiment of the present invention includes receiving a request for an original electronic document on a second network device (e.g., a content converter) on a first computer network from a first network device (e.g., a hand-held device) on the first computer network. Conversion information is extracted from the request. One or more conversion keys are created from the extracted conversion information to access one or more of multiple conversion preferences stored in a database. One or more of the multiple conversion preferences is obtained from the database using the one or more conversion keys. A conversion directive is created using conversion precedence to convert an original electronic document into a converted electronic document. The conversion directive includes a list of conversion preferences and uses conversion precedence indicating an ordering in which conversion preferences are to be applied to an original electronic document.

The conversion directive can be created from a number of different conversion preferences thereby providing a number of different conversion permutations. The conversion preferences include user-conversion preferences, device-conversion preferences, site-specific conversion preferences or other conversion preferences for converting an original electronic document into a converted electronic document suitable for display on a hand-held device or other desired device. The conversion directive can also be saved in persistent storage (e.g., a database) for later use.

In one preferred embodiment of the present invention, the conversion directive is used to convert most original electronic documents intended for viewing on a display with SuperVGA resolution into converted electronic documents based on one or more conversion preferences. A converted electronic document is in an appropriate format for a hand-held device or other device to display on a display with less than SuperVGA resolution. However,

the converted electronic document can also be converted for use on a device with a display with SuperVGA resolution.

A content conversion system for a preferred embodiment of the present invention includes a content converter, database, proxy server and content conversion directive. The content converter converts an original electronic document into a converted electronic document for a desired device based on a content conversion directive. The database stores multiple conversion preferences used for content conversion. The proxy server, communicates with a multiple network devices making requests for original electronic documents, communicates with a context converter, and with multiple content servers on a computer network providing original electronic documents. The content conversion directive with conversion precedence is created from one or more of the multiple conversion preferences from the database converts an original electronic document into a converted electronic document. The content conversion system components listed may also be split into additional components or combined into fewer components to provide content conversion and more or fewer system components can also be used.

The content conversion method and system allows user-conversion preferences, device-conversion preferences, site-specific conversion preferences and/or other conversion preferences to be selected to provide a customized converted electronic document that is suitable for display on a hand-held device or other devices. The content conversion method and system allow conversion of text, hypertext, images, as well as additional electronic document content (e.g., graphics, audio, video, etc.) for most original electronic documents obtained from electronic document servers on the World-Wide-Web on the Internet or an intranet.

The foregoing and other features and advantages of a preferred embodiment of the present invention will be more readily apparent from the following detailed description, with references to the accompanying drawings, in which:

FIG. 1 is a block diagram generally illustrating a content conversion system;

FIG. 2 is a block diagram illustrating a method for content conversion;

FIG. 3 is a block diagram illustrating one exemplary content conversion system;

FIGS. 4A, 4B and 4C are a flow diagram illustrating a method for content conversion using the content conversion system of FIG. 3;

FIG. 5 is a block diagram illustrating components of a content converter;

FIG. 6 is a flow diagram illustrating a method for content conversion with the content converter illustrated in FIG. 5;

FIG. 7 is a block diagram illustrating a method for content conversion using conversion preferences; and

FIGS. 8A and 8B are block diagrams illustrating exemplary content conversion screen displays.

Content Conversion System

FIG. 1 is a block diagram illustrating a content conversion system 10. Content conversion system 10 includes multiple components. A first network device 12 requests original electronic documents. The first network device 12 includes a display that is less that SuperVGA resolution (i.e., less than 800x600 pixel resolution or less than 256 colors). However, a preferred embodiment of the present invention can also be practiced with a first network device that has a display with SuperVGA resolution.

As is known in the art and is described above, an electronic document includes text, hypertext, graphical data or references to graphical data images, audio, video and other

content. A hypertext document includes markup codes called "tags." The structure of hypertext documents is defined by document markup languages such as Standard Generalized Markup Language ("SGML"), Hyper Text Markup Language ("HTML"), eXtensible Markup Language ("XML"), Virtual Reality Markup Language ("VRML") and others. Markup languages also allow references to additional content besides text including graphics, animation, audio, video and other electronic data.

Also described above, electronic documents are typically displayed for a user with a software application called a "browser." A browser on a hand-held device may be a sub-set of a larger browser, and not capable of displaying complete content of a requested electronic document as stored on an electronic document server. A browser typically reads an electronic document and renders the electronic document content into a visual display of text, graphics, animation, audio, video, etc., for display on a device such as a personal computer. Most electronic documents developed assume that users will view the content of the electronic document with a browser on a desktop computer screen with a standard "SuperVGA" resolution (e.g., 800x600 pixel resolution with 256 or more available colors). A user can alter display of selected content by changing browser attributes. However, the custom content for a user is limited by changeable attributes in the browser being used.

Returning to FIG.1, a proxy server 14 services requests for electronic documents from the first network device 12 and sends converted electronic documents back to the first network device 12. A content converter 16 converts content of an electronic document to a format usable on the first network device 12 (e.g., a device with a smaller display). A database 18 stores conversion preferences. FIG. 1 illustrates a single database 18. However, multiple database components can also be used. A computer network 20 provides one or more electronic document servers for supplying electronic documents.

In a preferred embodiment of the present invention, the computer network 20 is the World-Wide-Web on the Internet. As is known in the art, the Internet is a world-wide network of interconnected computers. The World-Wide-Web is an information system on the Internet designed for electronic document interchange. However, other computer networks with electronic document servers could also be used (e.g., an intranet).

FIG. 1 illustrates the content converter 16 as an individual component. However, the content converter 16 can also be integral to the proxy server 14 (not illustrated in FIG. 1). A preferred embodiment of the present invention is not limited to the network components shown in the content conversion system 10 and more or fewer network components may also be used. In a preferred embodiment of the present invention, the proxy server 14 and the content converter 16 are software components. In addition, the functionality of components from FIG. 1 can also be provided with a combination of hardware and software components, or hardware components.

In a preferred embodiment of the present invention, the functionality of components from FIG. 1 is provided with software using object-oriented programming techniques and the C++ programming language. However, other object-oriented programming languages besides C++ could also be used. In addition, in a preferred embodiment of the present invention, the functionality of components of FIG. 1 can also be provided with non-object oriented programming languages (e.g., C programming language).

As is known in the art, object-oriented programming is used to design computer software including object-oriented objects that are easy to create, cost effective to modify, and reusable. Object-oriented objects include "object data" and "object services." Object services are provided through "object methods" (also called "object operations" or "object functions"). Object methods typically operate on private data such as "instance data" or

"object state data" that an object owns. A collection of objects is called an "object class" which is sometimes called an "object type." An object class acts as a template that describes the behavior of sets of objects. An object's implementation is typically encapsulated, and is hidden from public view. Object private instance data can only be accessed by object methods of an object class. Object public instance data is accessed through a public "object interface."

An operating environment for components of content conversion system 10 of a preferred embodiment the present invention includes a processing system with at least one high speed Central Processing Unit ("CPU") and a memory system. In accordance with the practices of persons skilled in the art of computer programming, the present invention is described below with reference to acts and symbolic representations of operations that are performed by the processing system, unless indicated otherwise. Such acts and operations are referred to as being "computer-executed" or "CPU executed." Although described with one CPU, alternatively multiple CPUs may be used for a preferred embodiment of the present invention.

The memory system may include main memory and secondary storage. The main memory is high-speed random access memory ("RAM"). Main memory can include any additional or alternative high-speed memory device or memory circuitry. Secondary storage takes the form of persistent long term storage, such as Read Only Memory ("ROM"), optical or magnetic disks, organic memory or any other volatile or non-volatile mass storage system. Those skilled in the art will recognize that the memory system can comprise a variety and/or combination of alternative components.

Acts and symbolically represented operations include the manipulation of electrical signals by the CPU. The electrical signals cause transformation of data bits. The

maintenance of data bits at memory locations in a memory system thereby reconfigures or otherwise alters the CPU's operation. The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, optical, or organic properties corresponding to the data bits.

The data bits may also be maintained on a computer readable medium including magnetic disks, optical disks, organic disks and any other volatile or non-volatile mass storage system readable by the CPU. The computer readable medium includes cooperating or interconnected computer readable medium, which exist exclusively on the processing system or may be distributed among multiple interconnected processing systems that may be local or remote to the processing system.

Components of content conversion system 10 allow content conversion based on one or more sets of conversion preferences. An electronic document converted by the content conversion system 10 is suitable for display on a device with a smaller display and less than SuperVGA resolution or on a display with SuperVGA resolution.

Content Conversion

FIG. 2 is flow diagram illustrating a Method 22 for content conversion. At Step 24, the content converter 16 receives a request for an original electronic document from the first network device 12 via the proxy server 14. At Step 26, the content converter 16 consults the database 18 to obtain conversion preferences (e.g., conversion preferences to convert the original electronic document requested by the first network device 12 into a converted electronic document for the first network device 12). The conversion preferences can include any of user- preferences, device-conversion preferences, site-specific conversion preferences, or other preferences for content conversion. At Step 28, the content converter 16 returns the request for the original electronic document back to the proxy server 14. The proxy server 14

sends the request to an appropriate electronic document server on the computer network 20 to obtain the requested original electronic document.

At Step 30, the content converter 16 receives the requested original electronic document from the proxy server 14. At Step 32, content converter 16 converts the original requested electronic document into a converted electronic document based on conversion preferences (e.g., using a conversion directive or content conversion script) obtained at Step 26. Text, images and other hypertext content (e.g., audio, video, etc.) are converted based on one or more sets of preferences. At Step 34, the content converter 16 sends a converted electronic document to the proxy server 14. The proxy server 14 sends the converted electronic document to the first network device 12 in response to the request made by the first network device 12 for the original electronic document. The first network device 12 receives a converted electronic document based on one or more conversion preferences. Method 22 can also be practiced with the content converter 16 integral to the proxy server 14.

Method 22 allows a network device to receive a converted electronic document based on user-conversion preferences, device-conversion preferences, site-specific conversion preferences, or other preferences. Method 22 allows a personal digital assistant, hand-held computer, wireless phone, or other small or hand-held devices to receive a converted electronic document. The converted electronic document is appropriate for viewing on a smaller display with a lower resolution even though the original electronic document retrieved was written for a display in a higher resolution SuperVGA format. Method 22 can also be used for providing a converted electronic document for a display with SuperVGA resolution (e.g., by changing display characteristics based on one or more sets of conversion preferences).

Exemplary Content Conversion System

In one exemplary preferred embodiment of the present invention, the functionality of one or more components of the content conversion system 10 is split into multiple component modules and may include additional optional component modules (e.g., administrative modules, filter modules). Such an exemplary preferred embodiment is used to distribute functionality and provide additional conversion functionality to the content conversion system 10.

FIG. 3 is a block diagram illustrating an exemplary content conversion system 36 for one preferred embodiment of the present invention. However, the invention is not limited to the components illustrated in FIG. 3, and more or fewer components could also be used in the exemplary content conversion system. The original components of the content conversion system 10 from FIG. 1 are illustrated with a dashed line in FIG. 3. The original connections from FIG. 1 are not illustrated in FIG. 3. In addition, the conversion components could be split into more component modules or combined into fewer component modules.

An optional administrative interface 38 is used by system administrators with a second network device 40 to configure the conversion system 36 and to maintain the database 18. An optional administrative server 42 is a server used to modify configuration files used by the proxy server 14 and the database 18. A transaction manager 44 queries the database 18 for user-conversion preferences, device-preferences, site-specific conversion preferences, and other preferences during a request transaction.

A converter 46 communicates with client components 48, the content converter 16, and a cache 50, depending on the content type of an electronic document being converted. The converter 46 may also include a document production module (not illustrated in FIG. 3). The document production module can be use to help the converter 46 convert an electronic

document comprising a metadata object and a datapipe object into a converted electronic document.

The client components 48 include network components that retrieve electronic document content from the computer network 20. The cache 50 stores Uniform Resource Locators ("URLs"), request headers, response headers, converted data, unconverted data, and/or a conversion signature for request transactions. As is known in the art, a URL is used to locate an electronic document on the World-Wide-Web. Content conversion functionality provided by the proxy server 14 and the content converter 16 (FIG. 1) is distributed among additional component modules in the content conversion system 36 (FIG. 3). However, a preferred embodiment of the present invention can be practiced without distributing conversion functionality among the multiple modules illustrated in FIG. 3.

In a preferred embodiment of the present invention, the functionality of component modules from FIG. 3 are implemented with software using object-oriented programming techniques and the C++ programming language. However, other object-oriented programming languages and other non-object oriented programming languages could also be used. In addition, functionality of component modules from FIG. 3 can also be implemented as a combination of hardware and software components, or as hardware components.

Exemplary Content Conversion

FIGS. 4A, 4B and 4C are a flow diagram illustrating an exemplary Method 52 for content conversion using the content conversion system 36 of FIG. 3. At Step 54 of FIG. 4A, the proxy server 14 receives a request from the first network device 12 for an original electronic document. In one exemplary embodiment of the present invention, the request is a Hyper Text Transfer Protocol ("HTTP") request (e.g., http://www.spyglass.com/info.html).

However, other requests may also be made (e.g., File Transfer Protocol ("FTP") request, Gopher request, etc.) (e.g., ftp://www.spyglass.com/info.html). As is known in the art, HTTP is a transfer protocol used to transfer data from an electronic document server on the World-Wide-Web. For more information on HTTP see Internet Engineering Task Force ("IETF") Request For Comments ("RFC") 2068, incorporated herein by reference. File Transfer Protocol is a protocol that provides a way to access files on remote systems and is defined in RFC-172, incorporated herein by reference. Gopher is a protocol similar to File Transfer Protocol and provides a series of menus linked to files containing actual hypertext.

Returning to FIG. 4A at Step 56, the proxy server 14 which logs the request in a request log. At Step 58, a test is conducted by the proxy server 14 to determine if the first network device 12 is allowed to use content conversion functionality without login information. In a preferred embodiment of the present invention, the proxy server 14 uses network addresses (e.g., Internet Protocol ("IP")) addresses) for the test at Step 58. As is known in the art, IP is an addressing protocol designed to route traffic within a network or between networks. IP is described in RFC-791, incorporated herein by reference. However, other tests could also be used to determine if a network device is allowed to use content conversion without login information.

If a network device is not allowed to use content conversion without login information, at Step 60 the proxy server 14 prompts the first network device 12 for login information and logs a network address (e.g., IP address) for the first network device 12. If the first network device 12 is allowed to use content conversion functionality without login information at Step 58, or login information was obtained at Step 60, at Step 62 the proxy server 14 creates a object-oriented data structure called a "metadata" object and a "datapipe" object.

In an exemplary preferred embodiment of the present invention, a metadata object is a C++ object that conveys information such as request/response headers, conversion preferences and other information about a "databody" stored in a datapipe object. The datapipe object is also a C++ object. However, metadata objects and datapipe objects other than C++ objects could also be used. A databody is electronic document content such as hypertext markup languages (e.g., SGML, HTML, XML, VRML, etc.), text, graphical data, or graphics, animation, audio, video or other content that is stored in a datapipe object.

Table 1 illustrates an exemplary metadata object-oriented interface for a metadata object and an exemplary data structure for the metadata object. In an exemplary preferred embodiment of the present invention, object interfaces are Common Object Request Broker Architecture ("CORBA") interfaces. As is known in the art, CORBA is an architecture for creating, distributing and managing distributed program objects. However, other object architectures could also be used (e.g., Component Object Model ("COM") or Distributed Component Object Model ("DCOM") interfaces). Other object-oriented data structures could also be used and the present invention is not limited to the metadata object data structure illustrated in Table 1.

© by Spyglass, Inc.

```
// CORBA object interface for metadata object
typedef sequence<octect> CCMDSEQ;
interface CCMetaData
{
   void set (in string Category, in string Name, in string Value) //creates a new CCMetaData record
      raises (MDException); //error message in case of error
   void replace(in string Category, in string Name, in string Value) //replaces a value of a specified
      raises (MDException);                                          // CCMetaData record
   void get(in string Category, in string Name, out string Value)  //Gets a value of 1st occurrence of a
      raises (MDException);                                        // specified CCMetaDataRecord
   boolean exists(in string Category, in string Name)//Checks if a specified record exists
   void del (in string Category, in string Name) //Deletes 1st occurrence of a CCMetaData record
      raises (MDException);
   void getByRank(out string Category, in string Name, out string Value)//Gets a value of 1st occurrence
      raises (MDException);                                              //of a CCMetaData record
   void resetRank( ); //clears category rank settings and frees associated memory
      raises (MDException);
```

```
    void appendRank(in string Category)//Adds a specified request category to end of a rank list
      raises (MDException); //The 1st category appended has highest rank
    void signset(in string Category, in string Name, in string Value)//Adds a specified record to a
      raises (MDException); //conversion signature data object
    boolean SignatureValidate( ) //validate cache hits for a conversion signature
      raises (MDException);
    void serializedRequestMD(out CCMDSEQ serializedConvertedMD)//reserved for future use
      raises (MDException);
    void serializedSignatureMD(out CCMDSEQ SerializedSignatureMD)//reserved for future use
      raises (MDException);
    void debugRequestMD(in string FileName)//Outputs request-related metadata records
      raises (MDException);
    void debugResponseMD(in string FileName)//Outputs response-related metadata records
      raises (MDException);
    void debugSignatureMD(in string FileName)/Outputs signature-related metadata records
      raises (MDException);
    void impl_release( ); //release a ccmetadata record
    void impl_duplicate( ); //duplicate a ccmetadata record
};
```

Table 1.

Table 2 illustrates an exemplary object-oriented interface for a databody and an exemplary data structure for the databody. However, other object interfaces and data structures could also be used and the present invention is not limited to the databody interface and data structures illustrated in Table 2.

```
© by Spyglass, Inc.

typedef sequence <octect> CCDATASEQ;

//CORBA object interface for databody

interface CCOpData
{
        void impl_release();
        void impl_duplicate();
};

interface CCOpDataRaw : CCOpData
{
    attribute string name;  // name of the object
    attribute string type;  // type of the object
    attribute CCDATASEQ data; //CORBA sequence data for databody (e.g., hypertext)
};

//databody data structure
typedef struct _DATABODY
{
        char* pData;                    /* Pointer to data */
        size_t nSize;                   /* Size of body buffer */
        size_t nUsed;                   /* Bytes in use */
        size_t nMinGrow;                /* Minimum growth size */
```

```
} DATABODY;
```

Table 2.

Returning to FIG. 4A at Step 62, the proxy server 14 also creates a "datapipe" object-oriented object. The datapipe is an object-oriented object that can be read by mulitple readers, but written to by only one writer. In one preferred embodiment of the present invention, the datapipe object holds a databody including electronic document content.

Table 3 illustrates an exemplary object-oriented datapipe interface. However, other object interfaces could also be used and the present invention is not limited to the datapipe object illustrated in Table 3.

```
© by Spyglass, Inc.

// Datapipe buffer
typedef sequence<octet> dpBuf;

// Datapipe reader interface
//  Not thread safe
//          Threads should not share a reader, but have their own readers
interface dpReader
{
        // Release datapipe reader implementation
        void impl_release();

        // Read from datapipe asynchronously
        //  Returns immediately with octets currently available up to sizeDesired
        //  Returns TRUE until EOF read
        //  Caller must free returned dpBuf
        //  dpBuf only valid while datapipe or dpReader reference held by caller
        boolean readAsync(out dpBuf output, in unsigned long sizeDesired)
         raises(dpError);

        // Read from datapipe synchronously
        //  Blocks until sizeDesired octets read, EOF read, or abort signaled
        //  Returns TRUE until EOF read
        //  Caller must free returned dpBuf
        //  dpBuf only valid while datapipe or dpReader reference held by caller
        boolean read(out dpBuf output, in unsigned long sizeDesired)
         raises(dpError);

        // Read entire datapipe synchronously
        //  Blocks until all data available (EOF read), or abort signaled
        //  Caller must free returned dpBuf
        //  dpBuf only valid while datapipe or dpReader reference held by caller
        dpBuf readAll()
         raises(dpError);
```

```
        // Seek reader to new absolute file position
        //  Returns immediately
        //  Callers can seek to a desire file position before data is available
        //  Seeking past the EOF will cause an exception to be thrown on reading
        void seekAbs(in unsigned long filePos);

        // Seek reader to new relative file position
        //  Returns immediately
        //          Returns new file position
        //  Callers can seek to a desire file position before data is available
        //  Seeking past the EOF will cause an exception to be thrown on reading
        unsigned long seekRel(in long offset);
};

// Datapipe interface
interface datapipe
{
        // Duplicate datapipe
        //  Increments implementation reference count
        void impl_duplicate();

        // Release reference to datapipe
        //  Decrements implementation reference count
        //  The datapipe will be destroyed when the reference count reaches 0
        void impl_release();

        // Create a datapipe reader
        //  File position = 0
        dpReader createReader()
         raises(dpError);

        // Write input data to the end of the datapipe
        //  Data written is immediately available to all readers
        void write(in dpBuf input)
         raises(dpError);

        // Write end of file to datapipe
        //  Releases any unused datapipe memory
        //  All subsequent write operations will generate an exception
        void terminate();

        // Set abort signal on datapipe
        //  All subsequent read/write operations will generate an exception
        void abort();

        // Get bytes available for reading in datapipe
        unsigned long getAvailable();

        // Set datapipe size
        //  Changes the size of pipe after creation
        //  Intended for use when pipe initially sized to 0
        //          Note: will not size the pipe smaller than existing used size
        //                      does not affect file position
        void setSize(in unsigned long newSize)
         raises(dpError);
};

// Datapipe factory interface
```

```
interface dpFactory
{
        // Create a new datapipe object
        // Initial implementation reference count = 1
        // Specifying a minGrow of 0 causes the datapipe size to double for each grow
        datapipe create(in unsigned long initialSize, in unsigned long minGrow)
         raises(dpError);
};

interface CCOpDataPipe : CCOpData
{
        attribute string   name;
        attribute datapipe        pipe;
};
```

Table 3.

After creating the conversion objects, the proxy server 16 passes request/response data objects. For example, the proxy server 14 passes the request including the metadata object and the datapipe object to the transaction manager 44 through the datapipe object .

At Step 64, the transaction manager 44 adds user, device, site-specific, or other conversion preferences from the database 18 to the request metadata object. In a preferred embodiment of the present invention, the transaction manager 44 uses transfer protocol request headers to identify devices and an IP address and a user-name to identify a user. However, other identification methods could also be used. The transfer protocol request headers may include HTTP headers or Multipurpose Internet Mail Extension ("MIME") headers defined in RFC-2046, incorporated herein by reference.

At Step 66, the transaction manger 46 passes the request metadata object and datapipe object to the converter 46. At Step 68, the converter 46 creates a one or more conversion keys derived from the metadata object including user and device-identifiers and a site-identifier using a request URL. A first conversion key is used to query the cache 50 for a converted or unconverted databody that will satisfy the request. However, other information could also be used to create the conversion key. A second conversion key is used to obtain conversion preferences from the database 18.

In a preferred embodiment of the present invention, the first and second conversion keys do not include the same information. The first conversion key used to query the cache 50 includes a URL plus additional information from the metadata object. The second conversion key used to query the database 18 may include a user-agent name, a user-name, or a URL may be used to query a database other than database 18.

At Step 70 of FIG. 4B, a test is conducted by the converter 46 to determine if a "suitable" version of a converted databody for the request is already in the cache 50. In a preferred embodiment of the present invention, a suitable version of a converted databody is one that is not older than a predetermined date or time. However, other criteria could also be used to determine a suitable converted databody (e.g., criteria included in a conversion signature). If a suitable version of a converted databody exists in the cache 50, the converter 46 sends a datapipe object including the converted databody from the cache 50 to the transaction manager 44 at Step 72. If a suitable converted version of a databody for the request is not already in the cache 50 at Step 72, a test is conducted at Step 74 to determine if an original unconverted databody is already stored in the cache 50. If so, at Step 76 converter 46 sends a datapipe object including the unconverted databody from the cache 50 to the content converter 16 for conversion.

If an original unconverted databody is not stored in the cache 50 at Step 74, (i.e., the cache 50 does not contain any databody matching the request), the converter 46 sends the metadata object and datapipe object associated with the request to the content converter 16 at Step 78. The content converter 16 converts the request metadata object and datapipe object if necessary, and returns it to the converter 46 at Step 80.

At Step 82 of FIG. 4C, the converter 46 invokes the client components 48 to send the request to a desired electronic document server on the computer network 20 (e.g., World-

Wide-Web server on the Internet). At Step 84, the desired electronic document server on the computer network 20 returns an original electronic document to the converter 46 that is converted into a response databody in a response datapipe object and a response metatdata object (e.g., including response headers) by the client components 48.

At Step 86, the converter 46 saves the response databody as raw data and response metadata object including respsonse headers in the cache 50. Storing an unconverted response in the cache 50 may save transmission time when a network device makes a second request for an original electronic document, but with new conversion preferences. The unconverted response can then be used to create a converted response with new conversion preferences without contacting an electronic document again on the computer network 20.

At Step 88, the converter 46 sends the response databody stored in the response datapipe object to the content converter 16 for conversion. At Step 90, the content converter 16 performs any necessary conversions on the response metadata object and response databody in the response datapipe object and returns a converted response to the converter 46. At Step 92, the converter 46 sends a converted response including a converted databody in a datapipe object and a converted response metadata object to the cache 50 and to the transaction manager 44. Storing a converted response in the cache 50 saves processing time when a network device makes a second request for the same original electronic document with the same conversion preferences. A converted response can be returned without contacting an electronic document server again on the computer network 20.

At Step 94, the transaction manager 44 sends the converted response to the proxy server 14. At Step 96, the proxy server 14 sends the converted response to the first network device 12 via the user agent 38.

Exemplary Method 52 (FIG. 4) illustrates content conversion using the content conversion system 36 from FIG. 3. However, more or fewer steps can also be used with exemplary Method 52 and different content conversion system components could also be used. The exemplary preferred embodiment of the present invention is not limited to exemplary Method 52 or the content conversion system 36.

Exemplary Content Converter

In one exemplary preferred embodiment of the present invention, the content converter 16 is split into multiple component modules. FIG. 5 is a block diagram illustrating multiple components 108 of content converter 16. Content converter 16 includes a content conversion interface 110, a content conversion director 112, an operations module 114, an operation sequencer 116, a conversion script 118, a content conversion trader 120 and multiple conversion operations 122, 124, 126, three of which are illustrated. However, the content converter 16 can also be split into more or fewer component modules, and the exemplary preferred embodiment of the content converter 16 is not limited to the components illustrated in FIG. 5.

Content conversion interface 110 provides an interface between the content converter 16 and other components from the content conversion system 36 or the content conversion system 10. The content conversion director 112 directs content conversions. The operations module 114 is a library (e.g., a Dynamic Link Library) of functions that includes conversion operations. The operation sequencer 116 controls conversion operations including those listed in a conversion script 118. The conversion script 118 is a file that specifies which conversion operations should be executed for a selected content types (but not all content types) and in what order. The conversion script 118 includes a conversion directive. The content conversion trader 120 is a data reference registry for available conversion operations.

A first conversion operation can locate a second conversion operation by importing an object reference from the content conversion trader 120. The content conversion operations 122, 124, 126 include multiple conversion operations and reside in the operations module 114.

Tables 4 and 5 illustrate exemplary content conversion operations for the content conversion operators 122, 124, 126. However, more or fewer content conversion operations or conversions for other content types (e.g., VRML) could also be used. For example, another set of conversion operations may be used to convert HTML to an audio format (e.g., a wave format "*.WAV"), or convert an audio format to HTML).

Table 4 illustrates exemplary HTML conversion operations. However, more or fewer HTML conversion operations could also be used, and the conversion operations can also include other hypertext conversion operations (e.g., XML, SGML, VRML, etc.).

HTML Conversion Operation	Description
DOM_CREATE	Creates a Document Object Model ("DOM") from an HTML document..
DOM_EVAL	Evaluates one or more expressions that select one or more hypertext elements from a DOM.
DOM_TEMPLATE	Creates a new HTML document by inserting hypertext elements (e.g., selected by DOM_EVAL) into an HTML template.
HTML_ATTRIBUTE_ADD	Adds a specified attribute and value to instances of a specified tag.
HTML_ATTRIBUTE_MAX	Changes a value of a specified attribute to a maximum value in instances of a specified tag.
HTML_ATTRIBUTE_MIN	Changes a value of a specified attribute to a minimum value in instances of a specified tag.
HTML_ATTRIBUTE_REMOVE	Removes a specified attribute from instances of specified tag.
HTML_ATTRIBUTE_REPLACE	Replaces a value for instances of a specified tag.
HTML_ATTRIBUTE_SCALE	Scales a value for instances of a specified tag.
HTML_COMMENT_REMOVE	Remove comments from an HTML document.
HTML_ELEMENT_REMOVE	Removes instances of a specified HTML tag element.
HTML_IMAGE_ATTR_SCALE	Scales width and height attributes of instances of an HTML IMG tag.
HTML_IMAGE_TO_LINK	Removes an image from an HTML document and creates a link to the image.
HTML_IMAGEMAP_SCALE	Scales instances of HTML image maps.
HTML_SCALE_IMAGE_QUALITY	Scales an image to reduce its data size without changing the size of the image presented in an HTML document.
HTML_TABLE_TRANSLATE	Removes instances of table tags and reformats

	the contents of tables.
HTML_TAG_REMOVE	Removes instances of a specified tag.
HMTL_TAG_REPLACE	Replaces instances of a specified tag with a new tag.

Table 4.

Table 5 illustrates exemplary image conversion operations. However, more or fewer image conversion operations could also be used, and the conversion operations can also include other image conversion operations. The content converter 16 uses exemplary image conversion operators to convert images into an internal "raw image" format. The internal raw image format is used to quickly reduce and scale images, as well as convert one image format into another image format. However, the present invention is not limited to using a raw image format and other internal image formats could also be used.

Image Conversion Operation	Description
PI_REDUCE	Reduce an image to a specified color depth.
PI_SCALE	Scale an image.
PI_CONVERT_TO_GRAY	Convert a color image to a gray scale image.
BMP_TO_PI	Convert a Bit-MaPed ("BMP") image format to a raw image format.
JPEG_TO_PI	Convert a Joint Pictures Expert Group ("JPEG") image format to a raw image format.
PI_TO_BMP	Convert a raw image format to a BMP image format.
PI_TO_JPEG	Convert a raw image format to a JPEG image format.

Table 5.

FIG. 6 is a flow diagram illustrating a Method 128 for content conversion with the content converter 16 multiple components 108 illustrated in FIG. 5. At Step 130, the content converter 16 receives a conversion request from the converter 46 (e.g., Step 88 of FIG. 4C) or from the proxy server 14 (e.g., at Step 30 of FIG. 2) at the content conversion interface 110. At Step 132, the content conversion interface 110 supplies an object-oriented interface to a request metadata object and a request datapipe object including a databody that interact with other objects including the conversion operations 122, 124, 126. In a preferred embodiment of the present invention, the object interface is a CORBA interface called an "Interface Design Language" interface. However, other object interfaces could also be used (e.g.,

Component Object Model ("COM") or Distributed Component Object Model ("DCOM") interfaces). Table 6 illustrates an exemplary content conversion interface supplied by the content conversion interface 110.

```
© by Spyglass, Inc.

//Content Conversion Interface – Interface Design Language interface ("IDL")

module ContConv {

// Operation exception reason codes
enum opReason
{
        OP_FAILED,                      // Operation failed.
        OP_MEMERR,                      // Out of virtual memory.
        BAD_DATA,                       // CCOpData was wrong type, or had bad data.
        BAD_METADATA,                   // CCMetaData contained missing or bad data.
        BAD_ARG                         // Argument string was incomplete or bad format.
};

// Operation exception
exception opError
{
        opReason        reason;         // Reason code
        string          message;        // Descriptive message.
};


interface CCOperation
{
        void Do(inout CCOpData data,
                        inout CCMetaDataNS::CCMetaData metadata,
                        in string arg) raises (opError);
        void Restart();
};


};
```

Table 6.

After supplying the object interface content conversion interface, the content conversion interface 110 sends the conversion request to the content conversion director 112.

At Step 134, the content conversion director 112 calls the operation sequencer 116. In another embodiment of the present invention, a starting conversion component can also be specified in the request metadata object. By default, the operation sequencer 116 is the conversion component called to handle the conversion request. However, other content

converter 16 components 108 can also be used. At Step 136, the operation sequencer 116 reads the request metadata object and creates one or more conversion keys to determine a list of conversion operations that need to be called and a sequence order in which the conversion operations should be executed. In one preferred embodiment of the present invention, the list is a "conversion directive" with a "conversion precedence." In another embodiment of the present invention the conversion directive is stored in the conversion script 118. The list and sequences can also be obtained from information in the database 18.

In a sequence beginning at Step 138, for operations in the list of conversion operations, the operation sequencer 116 obtains an object address of a selected conversion operation (e.g., 122, 124, 126) from the content conversion trader 120. At Step 140, the selected conversion operation is executed. The executed conversion operation modifies the request databody stored in a datapipe object and request metadata object, if necessary and returns control to the object sequencer 116. The sequence including Steps 138 and 140 is repeated until all conversions desired by the conversion request are completed when control is returned to the content conversion director 112.

Table 7 illustrates exemplary source code to obtain an address for an object-oriented conversion method from the content conversion trader 120 and illustrates a generic object-oriented class for conversion operations. However, object-oriented methods code and other object-oriented classes could also be used. A name for a desired conversion operation is substituted for *OpName* in Table 7 (e.g., OpName = HTML_COMMENT_REMOVE, for removing comments from an HTML document).

© by Spyglass, Inc.

```
// creates an instance of Content Conversion Trader 120 ("CCTrader" )
CCTrader *pCCTrader = CCTrader::Instance( );
ContConv::CCOperation_var pOperation =
   PCCTrader->ImportOperation("operation_name"); //e.g., HTML_COMMENT_REMOVE
if(!CORBA::is_nil(pOperation))
```

```
  POperation->Do(data, metadata, arg);

//generic object-oriented class for a conversion operation
class OpName : public CCOperationIPL // e.g., class HMTL_COMMENT_REMOVE
(
   public:
      OpName(const char *OpName = "Internal_OpName"); //e.g., *OpName="spyga-comment-remove"
      Virtual ~OpName( ) { };
      Void Do(ContConv::CCOpData, *&data, CCMetaDataNS::CCMetaData *&metadata,
            Const char *arg);
};
```

Table 7.

At Step 142, the content conversion director 112 sends a converted metadata object and a converted databody in a datapipe object to the content conversion interface 110. At Step 144, content conversion interface returns the converted metadata object and converted databody in the datapipe object to the converter 46 (e.g., Step 90, FIG. 4C) or to the proxy server 14 (e.g., Step 34 of FIG. 2).

FIG. 5 and FIG. 6 illustrate one exemplary preferred embodiment of the present invention for content conversion in which the content converter 16 has been split into multiple components 108. However, this one exemplary preferred embodiment of the present invention is not limited to the multiple components illustrated in FIG. 5 or the steps illustrated with Method 128 of FIG. 6 and more or fewer conversion steps could also be used. In addition, the multiple content converter 16 components 108 of FIG. 5 can also be used for content converter 16 in content conversion system 10 illustrated in FIG. 1, or content conversion system illustrated in FIG. 3.

Content conversion using conversion preferences

In a preferred embodiment of the present invention, content conversion of an original electronic document is based on one or more conversion preferences. The conversion preferences are organized into multiple categories including user-conversion preferences, device-conversion preferences or site-specific conversion preferences or other conversion preferences. However, more or fewer conversion preferences could also be used.

After a network device with a smaller display requests an electronic document, the content converter 16 uses one or more sets of conversion preferences to determine how to convert the original electronic document into a converted electronic document usable by the network device.

FIG. 7 is a block diagram illustrating a Method 146 for content conversion using conversion preferences. At Step 148, content converter 16 receives a request for an original electronic document from the first network device 12 (e.g., via the proxy server 14 FIG. 1). The request includes a request metadata object with request headers (e.g., HTTP headers or MIME headers and IP headers), a request URL and other information and a datapipe object. At Step 150, conversion information is extracted from the request. In one exemplary preferred embodiment of the present invention, a user-identifier, a device-identifier, and a site-identifier based on a request URL are extracted from the request. However, other information and identifiers could also be extracted. In such an embodiment, transfer protocol headers from the request headers (e.g., HTTP headers or MIME headers) are used to identify a device-type and to identify a request URL. An IP address from a request header is used to identify a user location from the first network device 12 making the request. However, other information could also be used to identify a user, a device, or a site.

At Step 152, content converter 16 creates one or more conversion keys from the extracted conversion information to access one or more conversion preferences stored in the database 18. In one exemplary preferred embodiment of the present invention multiple conversion keys are created from a user-identifier, a device-identifier and site-identifier based on a request URL. However, other conversions keys could also be created (e.g., a user-identifier and device-identifier, a user-identifier and a site-identifier, or device-identifier and a site-identifier, etc.). In addition, access to the one or more conversion preferences could

also be from a source other than the database 18 (e.g., obtained from another server on another network or from another network device).

At Step 154, the conversion key to is used to obtain one or more conversion preferences for the request. In one exemplary preferred embodiment or the present invention user, device and site-specific (i.e., a URL) conversion preferences are obtained. However, more or fewer preferences could also be obtained. At Step 156, a conversion directive is created with conversion precedence from the obtained conversion preferences. In one preferred embodiment of the present invention, the conversion directive is stored in a metadata object. However, other conversion directive types and storage can also be used. The conversion directive is stored in persistent storage (e.g., the database 18, a user's hard drive, etc.) for later use. A number of different conversion directives can be created at Step 156, dependent on the conversion preferences being used including those from conversion script 118. Thus, a number of permutations are typically available to create a conversion directive since at least three types of conversion preferences are typically used (e.g., user, device and site-specific). Using more than three types of conversion preferences provides additional conversion permutations from which a conversion directive can be created.

The conversion directive based on selectable conversion preferences can be used to instruct the content converter 16 to convert an electronic document differently for different devices (e.g., a first device has a color display while a second device has a monochrome display; convert an electronic document differently for different users (e.g., a first user prefers to view images in a low resolution display mode, while another user desires to view images in a high resolution display mode); and convert an electronic document different for different sites (e.g., a first site may include electronic documents with a large number of images, while a second site may include only HTML documents with text and a few images.

The first site may require a user's device set one or more viewing parameters to view the images to be displayed).

At Step 158, the content converter 16 applies the conversion directive to convert an electronic document retrieved from an electronic document server on the computer network 20. The conversion precedence in the conversion directive provides a precedence hierarchy for content conversion.

In one exemplary embodiment of the present invention, user-conversion preferences are at a highest level in the precedence hierarchy, followed by device-conversion preferences, followed by site-specific conversion preferences. If user-conversion preferences are available, then they are used before device-conversion preferences or site-specific conversion preferences based on the exemplary hierarchy. If no user-conversion preferences are available, device-conversion preferences are used before site-specific conversion preferences based on the exemplary hierarchy. In another embodiment of the present invention, the precedence hierarchy includes different preference precedence (e.g., site-specific conversion preferences may have a higher precedence than user-conversion preferences based on the characteristics of a network site). In one preferred embodiment of the present invention, a preference hierarchy is statically assigned by a system administrator and cannot be changed by a user. In another embodiment of the present invention, the preference hierarchy may be dynamically changed by a user.

In cases where there is a conversion preference conflict, the content converter 16 determines which preferences have priority. An HTML site may require that a user's device set display resolution to high resolution mode to allow images to be adequately displayed. A site-specific conversion preference is then provided by the HTML site to set display resolution to high resolution. However, a user may not like the high resolution mode to be

used on the device, and provide a user-conversion preference for medium resolution mode. A conflict results, and the content converter 16 resolves the conflict. If user-conversion preferences have higher precedence, then the user-conversion preference (i.e., medium resolution) is used over the site-specific conversion preference. If site-specific conversion preferences have higher precedence, then the site-specific conversion preferences would be used over the user conversion preferences (i.e., high resolution).

As an example of the use of Method 146, the first network device 12, a Personal Digital Assistant ("PDA"), with an IP address of 128.10.20.30, requests a hypertext document with the URL "http://www.spyglass.com/info.html." Using Method 146, at Step 148 the content converter 16 receives a request for an original electronic document from the first network device 12 (e.g., via the proxy server 14 or the converter 46). The metadata object includes one or more request headers (e.g., HTTP or MIME) including an IP address of 128.10.20.30, a device-identifier of "PDA" and a URL component of "www.spyglass.com/info.html" and a datapipe object for a databody. At Step 150, conversion information is extracted from the request. A user-identifier of 128.10.20.30, a device-identifier of PDA and a site-identifier including request URL of "www.spyglass.com" is extracted from the request headers.

At Step 152, one or more conversion keys are created from the extracted information to access one or more conversion preferences stored in the database 18. In one exemplary preferred embodiment of the present invention, a first conversion key is created from the user-identifier 128.10.20.30 mapped to a user-name (e.g., "kenc"). A second conversion key is created from a device-identifier of PDA. A third conversion key is created with a site-identifier request URL "www.spyglass.com." However, more or fewer conversion keys could also be created.

At Step 154, the one or more conversion keys are used to obtain one or more conversion preferences for the request from the database 18. Table 8 illustrates exemplary preferences extracted from the database 18 for the request.

Type of Conversion Preference	Preferences
User ID = 128.10.20.30,kenc	Remove comments in hypertext documents. Change a display font to "Arial" in hypertext documents.
Site-Specific ID = www.spyglass.com	None.
Device ID = PDA	Convert images to grayscale. Scale images to 50% of original size.

Table 8.

Table 8 lists exemplary conversion preferences stored in non-precedence order in the database 18 so conversion precedence must be determined (i.e., user-converson preferences have highest precedence, device-conversion preferences have next highest preferences and site-specific conversion preferences have lowest precedence). However, in another preferred embodiment of the present invention, conversion preferences are stored in precedence order.

At Step 156, a conversion directive is created with conversion precedence from the obtained conversion preferences. An exemplary conversion directive for the conversion preferences shown in Table 8 is illustrated in Table 9. There are no conflicts to resolve in this example.

© by Spyglass, Inc. spyga_remove_comments:yes spyga_change_attribute:font=arial spyga_convert_to_gray:yes spyga_scale_image:scale=50%

Table 9.

Table 9 illustrates an exemplary conversion directive in precedence order. The user preferences are first (spyga_remove_comments, spyga_change_attribute:font="arial"), followed by the device preferences (spyga_convert_to_gray:yes, spyga_scale_image:scale="50%), followed by site-specific conversion preferences (i.e.,

none). The user preference to remove comments is executed before the user preference to change the display font to "Arial."

The conversion directive (e.g., Table 8) created at Step 156 of FIG. 7 is applied to an exemplary original electronic document at Step 158. An exemplary original HTML hypertext document is illustrated in Table 10. However, a conversion directive including the exemplary conversion directive from Table 9 can also be applied to other hypertext document types (e.g., XML, SGML or VRML).

```
<! Example.html>
<HTML>
<! This is an HTML comment>
<TITLE>Example</TITLE>
<BODY>
This is a conversion example.
<BR> <!new line>
<! Display a sample image>
<IMG src="http://www.spyglass.com/glasses.jpg">
<BR>
This is a sample image.
</BODY>
</HMTL>
```

Table 10.

The first network device 12 requests the hypertext document called "example.html" illustrated in Table 10 from the World-Wide-Web site "www.spyglass.com. Table 11 illustrates an exemplary converted electronic document after application of Step 158 of Method 146 (FIG. 7). HTML comments have been removed from the HTML document in Table 10 based on the first user preference illustrated in Table 9 (i.e., remove_comments).

```
<HTML>
<TITLE>Example</TITLE>
<FONT FACE="Arial">
<BODY>
This is a conversion example.
<BR>
<IMG src="http://www.spyglass.com/glasses.jpg">
<BR>
This is a sample image.
</BODY>
</HTML>
```

Table 11.

On line 3 of Table 11, the display font face has been changed to Arial with the HTML tag "<FONT FACE="Arial">". Note that this tag does not exist in Table 10. The image in Table 11, "glasses.jpg," is converted by the content converter 16 when the image is requested using conversion preferences store in the database 18. The conversion of the image is "invisible" to the user.

FIGS. 8A and 8B are block diagrams of screen displays 160, 162 illustrating content conversion with preferences. FIG. 8A illustrates a block diagram of a screen display 160 on a SuperVGA resolution monitor of the original electronic document from Table 10 before application of content conversion Method 146. The image "glasses.jpg" from Table 8 is a color image of sunglasses that are "red" in color. The default font for text displayed in FIG. 8A is a Times Roman font. The HTML document has several comments indicated by HTML comment tags "<!...>".

FIG. 8B illustrates a block diagram of a screen display 162 for a converted electronic document for a PDA first network device 12 from Table 11 after application of content conversion Method 146. The PDA display has less than SuperVGA resolution and is grayscale. The default display font for text displayed has been changed to Arial as is illustrated in FIG. 8B. The color image in the file "glasses.jpg" has been changed from color to grayscale and has been reduced in size by 50% by content converter 16. In FIG. 8A, a browser uses the HTML image tag "<IMG src="http://www.spyglass.com/glasses.jpg">" to display the red sunglasses image. In FIG. 8B, a browser in PDA first network device 12 parses the HTML image tag and requests the image. The content converter 16 converts the image based on conversion preferences stored in the database 18, and returns a converted image for display to the PDA first network 12 via the proxy server 14. The proxy server 14 returns a converted grayscale image reduced in size by 50%. The PDA first network device 12

receives a converted image as a result of requesting an original image without additional action by the PDA first network device.

If the requested "glasses image" was part of an image map, a converted HTML tag "<IMG src="http://www.spyglass.com/glasses_spyg$(spyga-image-attr-scale=50).jpg">" as a "virtual URL" is added to the HTML document in Table 11. For example, the virtual URL "glasses_spyg$(spyga-image-attr-scale=50)" within the HTML image tag allows an original image from an image map to be scaled to 50% of its original size with the virtual URL. The conversion information within the virtual URL text "image-name_spyg$(...).image_type" is used to scale the image to match the converted image map.

In a preferred embodiment of the present invention, the content converter 16 is closely associated with the proxy server 14. The proxy server 14 acts as an intermediary between a network device requesting an original electronic document and an electronic document server on a network providing an original electronic document. The proxy server 14 intercepts requests for hypertext documents and for hyperlinks within hypertext documents. As a result, when a browser on a user device parses the HTML image tag "<IMG src="http://www.spyglass.com/glasses.jpg">", the proxy server 14 provides a converted grayscale image reduce in size by 50% instead of the original image referenced in the original electronic document with the HTML by calling the content converter 16.

In a preferred embodiment of the present invention, a user sets desired user-conversion preferences by creating a user-conversion preference file on the proxy server 14. In another embodiment of the present invention, a user-conversion preference file is stored on a user device and retrieved by the content converter 16 or the proxy server 14. In yet another embodiment of the present invention, a user-conversion preference file may be stored on a specific site (e.g., on an Internet Service Provider site) and retrieved by the content converter

16 or the proxy server 14. Device-conversion preferences and site-specific preference files are typically created by a system administrator and not by user (e.g. with the administrative interface 38 of FIG. 3).

However, a user may override the device and site-specific conversion preferences as was discussed above, provided user-conversion preferences have the highest precedence in the conversion precedence scheme. In another embodiment of the present invention, a user is allowed to provide device-conversion preferences or site-specific conversion preferences.

The content conversion methods and systems described herein allow selection of conversion preferences to provide a customized converted electronic document that is appropriate for display on a hand-held device or other desired devices with displays with less than SuperVGA resolution. However, a converted electronic document can also be displayed on a display with SuperVGA resolution. Text, images, as well as additional electronic document content (e.g., hypertext tags hypertext tag parameters, audio video, etc.) are converted for most original electronic documents obtained with SuperVGA resolution from content servers on the World-Wide-Web via the Internet or an intranet. The content conversion methods and systems described herein provide flexibility for new devices and are easily extended by adding new content conversion operation modules for new content that may become available.

In view of the wide variety of embodiments to which the principles of the present invention can be applied, it should be understood that the illustrated embodiments are exemplary only, and should not be taken as limiting the scope of the present invention. For example, the steps of the flow diagrams may be taken in sequences other than those described, and more or fewer elements may be used in the block diagrams. The claims should not be read as limited to the described order or elements unless explicitly stated to that effect.

CLAIMS

1. A method of content conversion in a first computer network with a plurality of network devices connected to a second computer network with a plurality of network devices, comprising the following steps:

receiving a request for an original electronic document on a second network device on the first computer network from a first network device on the first computer network;

extracting conversion information from the request;

creating one or more conversion keys from the extracted conversion information to access one or more of a plurality of conversion preferences stored in a database;

obtaining one or more of the plurality of conversion preferences from the database using the one or more conversion keys; and

creating a conversion directive with a conversion precedence to convert an original electronic document into a converted electronic document.

2. A method as claimed in Claim 1 further comprising:

sending the request for the original electronic document from the second network device to a third network device on the second network;

receiving a requested original electronic document on the second network device from the third network device;

converting the requested original electronic document using the conversion directive, thereby creating a converted electronic document; and

sending the converted electronic document from the second network device to the first network device.

3. A method as claimed in Claim 1 or 2, further comprising storing the conversion directive in persistent storage for later use.

4. A method as claimed in Claim 1, 2 or 3, wherein the conversion precedence includes from highest-to-lowest precedence, user-conversion preferences, device-conversion preferences and site-specific conversion preferences.

5. A method as claimed in any preceding claim, wherein the one or more conversion keys include individual conversion keys to obtain user-conversion preferences, device-conversion preferences and site-specific conversion preferences.

6. A method as claimed in any preceding claim, wherein the conversion directive includes a list of conversion preferences and conversion precedence indicating an ordering in which the list of conversion preferences are to be applied to an original electronic document.

7. A method as claimed in any preceding claim, wherein the original electronic document is a Hyper Text Markup Language document or an Extensible Markup Language document.

8. A method as claimed in any preceding claim, wherein the first network device is a hand-held device and the second network device is a content converter.

9. A method as claimed in any preceding claim, wherein the step of obtaining one or more of the plurality of conversion preferences includes obtaining conversion preferences

for: converting a hypertext document into a document object model; adding a specified attribute to instances of a specified hypertext tag; changing a value of a specified attribute to a maximum value in instances of a specified hypertext tag; changing a value of a specified attribute to a minimum value in instances of a specified hypertext tag; removing a specified attribute from instances of a specified hypertext tag; replacing values of a specified attribute in instances of a specified hypertext tag; scaling a value of a specified attribute in instances of a specified hypertext tag; removing instances of a specified attribute in instances of a specified hypertext tag; scaling width and height of an image attribute in instances of a specified hypertext tag; removing a reference to an image and replacing the reference with a hyperlink to the image in a hypertext image tag; scaling instances of an image map in a hypertext tag; scaling an image to reduce its data size without changing the size of the image; removing hypertext table tags and reformatting the contents of tables; removing instances of a specified hypertext tag; or replacing instances of a specified hypertext tag with a different hypertext tag for a hypertext document.

10. A method as claimed in any preceding claim, wherein the step of obtaining one or more of the plurality of conversion preferences from the database using the conversion key includes obtaining image conversion preferences for: reducing an image to a specified color depth; scaling an image; converting a color image to a gray scale image; converting a bit-map format to a raw image format; converting a joint pictures expert group format to a raw image format; converting a raw image format to a bit-map format; or converting a raw image format into a joint pictures expert group format for an image.

11. A method of content conversion in a first computer network with a plurality of network devices connected to a second computer network with a plurality of network devices, comprising the following steps:

receiving an original electronic document on a second network device on the first network from a third network device on the second network;

obtaining a conversion directive with a conversion precedence on the second network device, wherein the conversion directive includes a list of conversion preferences and uses conversion precedence indicating an ordering in which the list of conversion preferences are to be applied to the original electronic document; and

converting the original electronic document into a converted electronic document using the conversion directive.

12. A method as claimed in Claim 11, further comprising sending the converted electronic document to a first network device on the first network, wherein the first network device displays the converted electronic document.

13. A method as claimed in Claim 11 or 12 wherein the conversion precedence includes user- conversion preferences, device-conversion preferences and site-specific conversion preferences.

14. A method as claimed in Claim 11, 12 or 13, wherein the converted electronic document is suitable for display on a hand-held device with less than SuperVGA resolution.

15. A method as claimed in any of Claims 11 to 14, wherein the first network device is a hand-held device and the second network device is a content converter.

16. Computer software adapted when run in data processing means to carry out a method as claimed in any preceding claim.

17. Computer software as claimed in claim16, recorded on a computer readable medium.

18. A content conversion system comprising, in combination:

a content converter for converting an original electronic document into a converted electronic document for a desired device based on a content conversion directive;

a database for storing a plurality of conversion preferences used for content conversion;

a proxy server for communicating with a plurality of network devices making requests for original electronic documents, communicating with the context converter, and with a plurality of content servers on a computer network providing original electronic documents; and

a content conversion directive with a conversion precedence created from one or more of the plurality of conversion preferences from the database for converting an original electronic document into a converted electronic document.

19. A system as claimed in Claim 18, wherein the plurality of conversion preferences include any of: user-conversion preferences, device-conversion preferences or site-specific conversion preferences.

20. A system as claimed in Claim 18 or 19, wherein the database for storing a plurality of conversion preferences used for content conversion includes storing conversion preferences for: converting a hypertext document to a document object model; adding a specified attribute to instances of a specified hypertext tag; changing a value of a specified attribute to a maximum value in instances of a specified hypertext tag; changing a value of a specified attribute to a minimum value in instances of a specified hypertext tag; removing a specified attribute from instances of a specified hypertext tag; replacing values of a specified attribute in instances of a specified hypertext tag; scaling a value of a specified attribute in instances of a specified hypertext tag; removing instances of a specified attribute in instances of a specified hypertext tag; scaling width and height of an image attribute in instances of a specified hypertext tag; removing a reference to an image and replacing the reference with a hyperlink to the image in a hypertext image tag; scaling instances of an image map in a hypertext tag; scaling an image to reduce its data size without changing the size of the image; removing hypertext table tags and reformatting the contents of tables; removing instances of a specified hypertext tag; or replacing instances of a specified hypertext tag with a different hypertext tag conversion preferences; reducing an image to a specified color depth; scaling an image; converting a color image to a gray scale image; converting a bit-map format to a raw image format; converting a joint pictures expert group format to a raw image format; converting a raw image format to a bit-map format; or converting a raw image format into a joint pictures expert group format.

21. Computer software which when run on data processing means will cause such data processing means to operate as a system as claimed in Claim 18, 19 or 20.

22. Computer software as claimed in Claim 21, recorded on a computer readable medium.

23. A method of content conversion, substantially as hereinbefore described with reference to the accompanying drawings.

24. A content conversion system , substantially as hereinbefore described with reference to the accompanying drawings.

25. Computer software, substantially as hereinbefore described with reference to the accompanying drawings.

The Patent Office

48

INVESTOR IN PEOPLE

Application No: GB 9918517.5 **Examiner:** Julyan Elbro
Claims searched: 1 to 25 **Date of search:** 27 March 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): G4A (AFD, AUDB)

Int Cl (Ed.7): G06F 17/21, 17/30

Other: ONLINE: COMPUTER EPODOC JAPIO WPI Internet

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
E, X	GB 2339374 A	IBM see abstract, figs. 1 and 6, page 1 line 15, and claim 1.	1, 2, 6 to 12, 14 to 18, 20, 21, 22
P, X	GB 2329310 A	IBM see abstract, page 21 lines 8 and 9, and page 25 lines 12 and 13.	11, 12, 16, 17, 18, 21, 22
E, X	EP 0947935 A2	IBM see abstract and fig. 5A.	11, 12, 16, 17, 18, 21, 22
P, X	EP 0926610 A1	INFOSIS GROUP see abstract.	1, 2, 6, 11, 12, 16, 17, 18, 21, 22
X	EP 0778534 A1	SUN MICROSYSTEMS see abstract, page 5 lines 24 to 31 and 48 to 53, page 6 line 11, page 7 lines 27 to 54.	1 to 9, 11 to 22
E, X	WO 99/57657 A1	LEXTRON SYSTEMS see abstract, fig. 8, and page 7 lines 24 and 25.	1, 2, 6, 7, 8, 11, 12, 14 to 18, 21, 22

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

The Patent Office

INVESTOR IN PEOPLE

Application No: GB 9918517.5 **Examiner:** Julyan Elbro
Claims searched: 1 to 25 **Date of search:** 27 March 2000

Category	Identity of document and relevant passage		Relevant to claims
P, X	WO 99/35802 A2	MICROSOFT see abstract, fig. 1, page 1 lines 11 and 12, and page 5 lines 5 to 30.	1, 2, 6, 11, 12, 16, 17
P, X	WO 98/45793 A1	TECHWAVE see abstract and fig. 4.	11, 12, 16, 17, 18, 21, 22
X	WO 97/18635 A2	IBM see abstract and claims 1, 7, and 8.	1, 2, 6, 11, 12, 16, 17, 18, 21, 22
X	WO 97/18516 A1	COMPUSERVE see abstract, page 9 lines 11 and 12, and fig. 9.	1, 2, 6, 7, 11, 12, 16, 17, 18, 21, 22
X	WO 97/17662 A1	C/NET see abstract and claims 1 and 2.	1, 2, 6, 8, 11, 12, 14 to 18, 21, 22
X	US 5742762	TELOGY NETWORKS see abstract.	11, 12, 16, 17, 18, 21, 22

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.